

LUIGI LO RUSSO
ELENA BIANCHI

SISTEMI E RETI

Per l'articolazione **INFORMATICA**
degli Istituti Tecnici
settore Tecnologico

2



HOEPLI

Sistemi e reti

LUIGI LO RUSSO ELENA BIANCHI

Sistemi e reti

Per l'articolazione Informatica degli Istituti Tecnici settore Tecnologico

VOLUME 2



EDITORE ULRICO HOEPLI MILANO



UN TESTO PIÙ RICCO E SEMPRE AGGIORNATO
Nel sito **www.hoepliscuola.it** sono disponibili:
- materiali didattici integrativi;
- eventuali **aggiornamenti** dei contenuti del testo.

Via Hoepli 5, 20121 Milano (Italy)
tel. +39 02 864871 – fax +39 02 8052886
e-mail hoepli@hoepli.it

www.hoepli.it



Indice

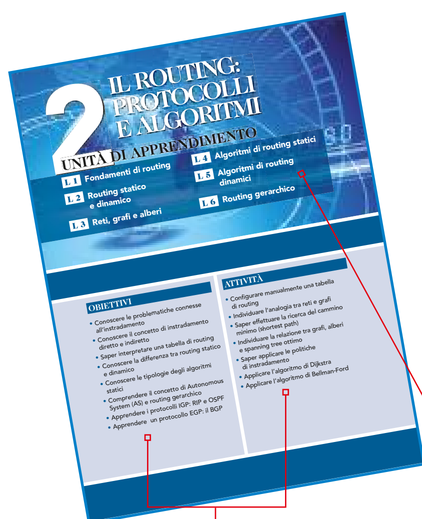
Presentazione

L'impostazione del presente corso in tre volumi è stata realizzata sulla base delle indicazioni ministeriali in merito a conoscenze e abilità proposte per la nuova disciplina Sistemi e Reti. L'opera è in particolare adatta all'articolazione Informatica degli Istituti Tecnici settore Tecnologico, dove la materia è prevista nel secondo biennio e nel quinto anno del nuovo ordinamento.

Abbiamo ritenuto irrinunciabile fare tesoro della nostra esperienza maturata nel corso di numerosi anni di insegnamento che ci ha reso consapevoli della difficoltà di adeguare le metodologie didattiche alle dinamiche dell'apprendimento giovanile e ai continui cambiamenti tecnologici che implicano sempre nuove metodologie di comunicazione, per proporre un testo con una struttura innovativa, riducendo l'aspetto teorico e proponendo un approccio didattico di apprendimento operativo, privilegiando il "saper fare".

Il testo, arricchito di contenuti che lo rendono di facile lettura, grazie ai richiami a vocaboli nuovi, spesso in lingua inglese, e ad ampie sezioni di approfondimento, aiuta lo studente a una maggior comprensione degli argomenti, trattati fino ad oggi in modo assai nozionistico. Inoltre, le schede per il laboratorio rappresentano un valido strumento per il rafforzamento dei concetti assimilati attraverso esercitazioni operative.

Il secondo volume è strutturato in cinque unità di apprendimento suddivise in lezioni che ricalcano le indicazioni dei programmi ministeriali per il quarto anno di studio: lo scopo di ciascuna unità di apprendimento è quello di presentare un intero argomento, mentre quello delle lezioni è quello di esporne un singolo aspetto.

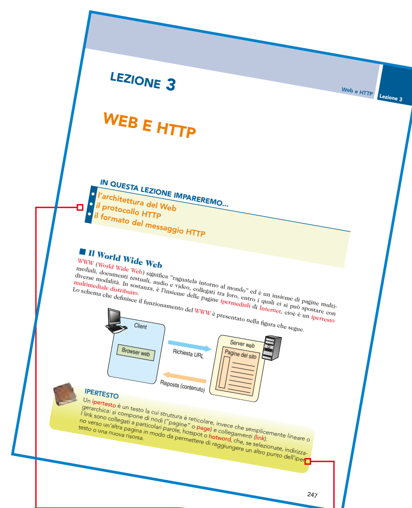

2 IL ROUTING: PROTOCOLLI E ALGORITMI

UNITÀ DI APPRENDIMENTO

L 1 Fondamenti di routing
L 2 Routing statico e dinamico
L 3 Reti, grafi e alberi
L 4 Algoritmi di routing statici
L 5 Algoritmi di routing dinamici
L 6 Routing gerarchico

OBIETTIVI
- Conoscere le problematiche connesse all'instradamento
- Conoscere il concetto di instradamento diretto e indiretto
- Saper interpretare una tabella di routing
- Conoscere la differenza tra routing statico e dinamico
- Conoscere le tipologie degli algoritmi statici
- Comprendere il concetto di Autonomous System (AS) e routing gerarchico
- Apprendere i protocolli IGP: RIP e OSPF
- Apprendere un protocollo EGP: il BGP

ATTIVITÀ
- Configurare manualmente una tabella di routing
- Individuare l'analogia tra reti e grafi
- Saper effettuare la ricerca del cammino minimo (shortest path)
- Individuare la relazione tra grafi, alberi e spanning tree ottimo
- Saper applicare le politiche di instradamento
- Applicare l'algoritmo di Dijkstra
- Applicare l'algoritmo di Bellman-Ford


Web e HTTP | Lezione 3

LEZIONE 3

WEB E HTTP

IN QUESTA LEZIONE IMPAREREMO...
- l'architettura del Web
- il protocollo HTTP
- il formato del messaggio HTTP

■ Il World Wide Web

WWW (World Wide Web) significa "ragnatela intorno al mondo" ed è un insieme di pagine multimediali, documenti testuali, audio e video, collegati tra loro, entro i quali ci si può spostare con diverse modalità. In sostanza, è l'insieme delle pagine ipermediali di Internet, cioè è un ipertesto multimediale distribuito.
Lo schema che definisce il funzionamento del WWW è presentato nella figura che segue.

IPERTESTO
Un ipertesto è un testo la cui struttura è reticolare, invece che semplicemente lineare o gerarchica: si compone di nodi ("pagine" o page) e collegamenti (link). I link sono collegati a particolari parole, hotspot o hotword, che, se selezionate, indirizzano verso un'altra pagina in modo da permettere di raggiungere un altro punto dell'iper testo o una nuova risorsa.

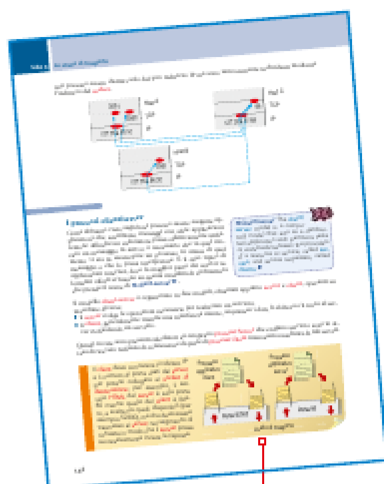
247

Nella pagina iniziale di ogni unità di apprendimento è presente un indice delle lezioni trattate

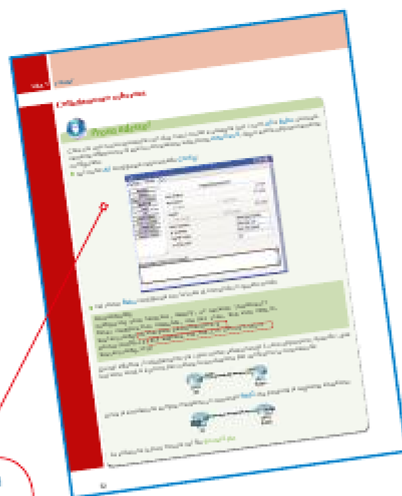
Indice degli obiettivi che si intendono raggiungere e delle attività che si sarà in grado di svolgere

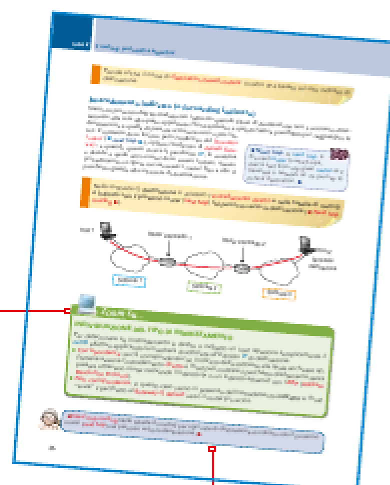
All'inizio di ogni lezione sono indicati in modo sintetico i contenuti

Il significato di moltissimi termini informatici viene illustrato e approfondito



Le finalità e i contenuti dei diversi argomenti affrontati sono presentati all'inizio di ogni unità di apprendimento; in conclusione di ogni lezione sono presenti esercizi di valutazione delle conoscenze e delle competenze raggiunte, suddivisi in domande a risposta multipla, vero o falso, a completamento, ed infine esercizi di progettazione da svolgere autonomamente.





Le osservazioni aiutano lo studente a comprendere e ad approfondire

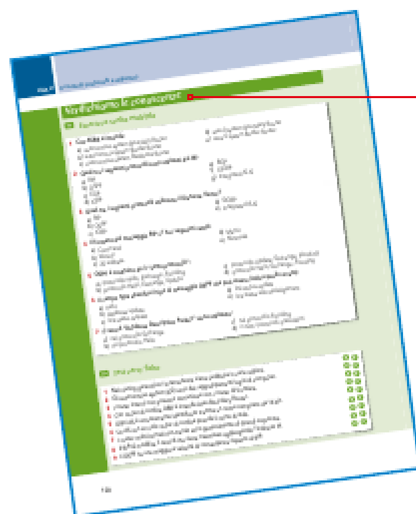
Lo studente può mettere in pratica in itinere quanto sta apprendendo nel corso della lezione

In questa sezione viene approfondito un argomento di particolare importanza

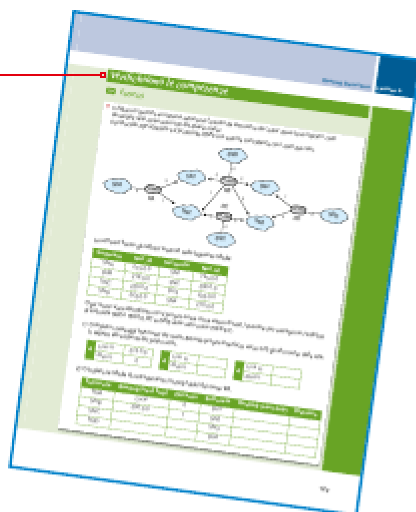
Le parole chiave vengono poste in evidenza e spiegate allo studente



Alla pagina web http://www.hoepliscuola.it sono disponibili le risorse online, tra cui unità didattiche integrative, numerosi esercizi aggiuntivi per il recupero e il rinforzo, nonché schede di valutazione di fine unità.



Per la verifica delle conoscenze e delle competenze è presente un'ampia sezione di esercizi



1 I ROUTER

UNITÀ DI APPRENDIMENTO

OBIETTIVI

- Conoscere l'architettura di un router
- Riconoscere i componenti hardware di un router
- Conoscere le funzionalità di un router
- Apprendere le caratteristiche di un SO per i router
- Conoscere la procedura di boot
- Conoscere la gerarchia dei comandi IOS

ATTIVITÀ

- Aggiungere interfacce a un router
- Cambiare modalità operativa in un router
- Configurare un router
- Utilizzare l'interfaccia CLI di IOS
- Inserire comandi nelle diverse modalità di accesso
- Utilizzare i router con Packet Tracer
- Connettere due router in seriale
- Connettere due router in ethernet

LEZIONE 1

ARCHITETTURA HARDWARE DI UN ROUTER

IN QUESTA LEZIONE IMPAREREMO...

- i compiti del router
- l'architettura di un router
- l'utilizzo di un router in Packet Tracer

Generalità

La comunicazione tra dispositivi e mezzi eterogenei viene resa possibile da strutture organizzative e protocolli di comunicazione appositamente studiati e realizzati, che definiscono "il linguaggio comune" per fare dialogare ogni device (ricordiamo tra tutti il protocollo TCP-IP).



RETE DI COMPUTER
Una rete di computer è un insieme eterogeneo di dispositivi che mediante apposite schede di interfacciamento (schede di rete) comunicano tra loro con diverse tipologie di mezzi di comunicazione, dal cavo telefonico alla fibra ottica, dalla comunicazione WI-FI al satellite, trasferendosi dati.

I protocolli di rete hanno la caratteristica di avere una struttura a livelli sovrapposti e i protocolli definiscono le modalità di comunicazione sia all'interno dei singoli livelli che tra livelli adiacenti, sia sottostanti che a livello superiore.

Il modello OSI (Open Systems Interconnection) standardizzato nel 1984 dalla ISO (International Organization for Standardization) è strutturato in 7 livelli ed è il pioniere del TCP-IP, il modello utilizzato oggi in Internet, composto da soli 4 livelli, con la corrispondenza con il modello ISO-OSI riportata in figura.

ISO-OSI	TCP-IP	Nome	Esempi
Livello 7	Livello 4	Application (Applicazione)	Telnet, FTP, SMTP ecc.
Livello 6			
Livello 5	Livello 3	Transport (Trasporto)	TCP, UDP
Livello 4			
Livello 3	Livello 2	Network (Rete)	IP (ICMP, IGMP)
Livello 2	Livello 1	Link (Collegamento)	NIC e relativi drivers
Livello 1			

Le reti di computer sono costituite da macchine che interagiscono per condividere e scambiare dati tra applicazioni software, operanti su computer dislocati fisicamente sia all'interno della stessa stanza o dello stesso edificio o residenti in uffici posti "all'altro capo della Terra".

I computer sono quindi appartenenti o a reti locali (LAN) o ad ampie aree geografiche (reti WAN e GAN); la connessione di queste reti tra loro prende il nome di ◀ **Internetworking** ▶ e viene realizzata per massimizzare l'efficienza e le performance garantendo affidabilità e sicurezza nelle comunicazioni.



◀ **Internetworking** (a combination of the words inter, "between", and networking) is the practice of connecting a computer network with other networks through the use of gateways that provide a common method of routing information packets between the networks. The resulting system of interconnected networks is called an internetwork, or simply an internet. ▶

Tra le apparecchiature utilizzate nell'internetworking fondamentale importanza hanno i router, dispositivi che, oltre a permettere l'instradamento dei dati, la connessione tra reti WAN e a Internet, vengono utilizzati per effettuare la segmentazione logica delle rete, in modo da ottenere maggiori prestazioni limitando i broadcast alle singole subnet, migliorando la sicurezza con regole di accesso per gli utenti alle singole subnet, facilitando l'amministrazione delle stesse (la segmentazione si ispira al concetto del *divide et impera*, secondo il quale è più semplice gestire piccole organizzazioni piuttosto che grandi reti).

L'instradamento (routing) è una funzione del livello 3 del modello OSI che, grazie a un'organizzazione gerarchica, permette di raggruppare insiemi di indirizzi e gestirli come una sola destinazione e individua il percorso più efficiente affinché il messaggio inviato dal computer mittente possa raggiungere il destinatario.

Il router è il principale dispositivo che attua il routing, svolgendo due funzioni fondamentali:

1. creare e gestire le tabelle di routing (routing table) e comunicare mediante opportuni protocolli (routing protocol) ogni loro variazione agli altri router in modo che possano mantenere aggiornata la topologia della rete;
2. determinare il percorso sul quale istradare un pacchetto utilizzando le routing table: in base al destinatario il router lo pone su una delle sue interfacce, quella alla quale è connesso il prossimo passo (next hop) che lo avvicina alla meta.

Il router

Il router è un dispositivo che ha le stesse componenti base di un PC (CPU, memoria, system bus e interfacce input/output) e come un PC può essere di diverse forme, dimensioni, con caratteristiche e performance differenti a seconda del suo impiego che può essere dalla gestione di una piccola rete domestica con connessione WI-FI alle medie o grandi reti aziendali o geografiche.

La figura a lato riporta un router di medie dimensioni a confronto con un router domestico. ▶



I router sono computer dedicati, progettati per operare in un determinato ambito con compiti specifici dove non sono adatti i PC commerciali, ma come ogni PC necessitano di un sistema operativo per il supporto delle applicazioni denominato IOS (Internetwork Operating System).

Come ogni sistema operativo, l'IOS, che sarà descritto nella prossima lezione, si occupa del funzionamento e della gestione del processore e dell'interprete di comandi per comunicare con l'utente: supporta i file di configurazione contenenti i dati necessari per effettuare il setup dei protocolli di rete e dei protocolli di routing abilitati sul router.

I router vengono opportunamente configurati dall'amministratore della rete in modo da individuare il cammino che devono percorrere i pacchetti verso la loro destinazione (routing statico o dinamico): in sintesi effettuano una commutazione trasferendo i pacchetti che gli arrivano in ingresso sulla corretta interfaccia d'uscita (switching).

I componenti principali interni di un router possono essere raggruppati in quattro gruppi:

1. porte di ingresso: sono in numero di m ≥ 2;
2. porte di uscita: sono in numero di m ≥ 2;
3. matrice di commutazione: collega ciascuna porta di ingresso a ciascuna porta di uscita;
4. processore di controllo: con la RAM, la NVRAM, la flash memory, la ROM esegue l'algoritmo di instradamento memorizzando la tabella di instradamento e aggiorna le tabelle di inoltro residenti in ciascuna delle porte di ingresso.

Router Cisco 2600 Series

Come esempio di router descriviamo la serie di router Cisco 2600 che è di fascia ◀ **midrange** ▶ e appartiene alla famiglia dei Modular Access Router: tra le principali caratteristiche la più importante è proprio la struttura modulare che permette di installare su di essi praticamente qualsiasi tipo di interfaccia di rete.



◀ **Midrange** Midrange computers, or midrange systems, are a class of computer systems which fall in between mainframe computers and microcomputers. ▶

L'espansione delle schede aggiuntive viene fatta inserendo le interfacce in due tipi di slot:

- Network Module Slot: dove vengono aggiunte le porte di tipo Ethernet o Token Ring;
- WAN Interface Card Slot: dove vengono aggiunte le porte seriali o ISDN.

Sono inoltre dotati di un "Advanced Integration Module Slot" dove poter aggiungere hardware specifico per migliorare le prestazioni della CPU (per esempio per aggiungere un coprocessore o un modulo di compressione hardware).

Il datasheet è scaricabile all'indirizzo http://www.cisco.com/web/IT/solutions/smb/pdf/net_found/2600_ds.pdf oppure nella sezione materiali del sito http://www.hoepliscuola.it dedicata a questo volume.

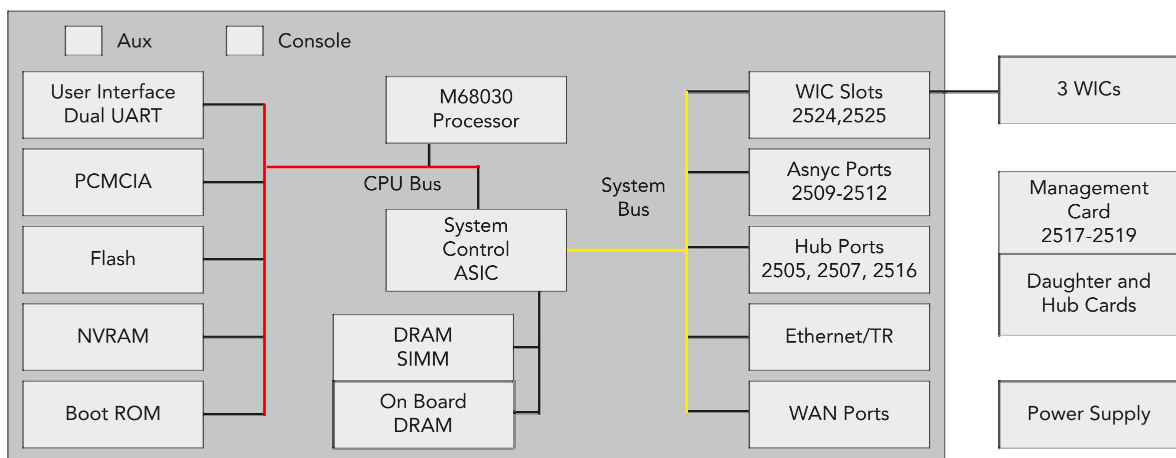


◀ Datasheets Il termine "datasheets" (letteralmente "schede tecniche") sta a indicare il manuale tecnico di un dispositivo o di un circuito che oltre alla specifica del componente ne riporta le informazioni funzionali e tecniche: nel caso di un circuito integrato ne riporta lo schema interno, le curve del comportamento in funzione della corrente, temperatura, e il packaging (imballaggio) con la descrizione degli ingressi/uscite, della piedinatura o delle porte oltre alle dimensioni fisiche del dispositivo stesso. ▶

La seguente tabella riporta le caratteristiche di ogni dispositivo della famiglia 2600:

Platform	NMs	AIMs	WICs	Fixed LAN Ports	Perfor-mance (Up to Kpps)	DRAM (Default MB/ Max MB)	FLASH (Default MB/ Max MB)	Included IOS Feature Set
CISCO2610/11	1	1	2	1E/2E	15	32/64	8/16	IOS IP
CISCO2610XM/11XM	1	1	2	1FE/2FE	20	32/128	16/48	IOS IP
CISCO2612	1	1	2	1TR/1E	15	32/64	8/16	IOS IP
CISCO2620/21	1	1	2	1FE/2FE	25	32/64	8/32	IOS IP
CISCO2620XM/21XM	1	1	2	1FE/2FE	30	32/128	16/48	IOS IP
CISCO2650/51	1	1	2	1FE/2FE	37	32	8/32	IOS IP
CISCO2650XM/51XM	1	1	2	1FE/2FE	40	64/128	16/48	IOS IP
CISCO2691	1	2	3	2FE	70	64/256	32/128	IOS IP

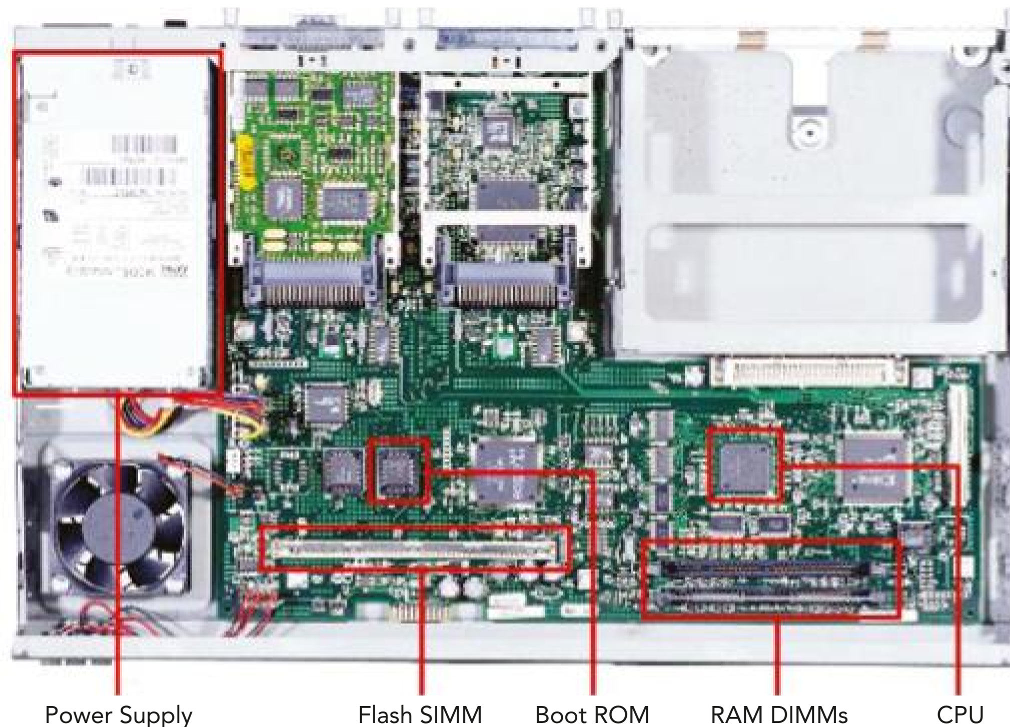
L’architettura di un router Cisco della serie 2600 è la seguente:



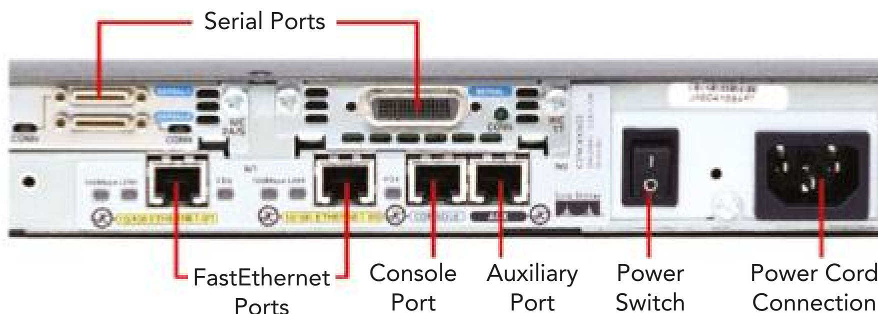
Descriviamo sinteticamente i componenti hardware dato che sono quelli di un normale personal computer:

- CPU: è il microprocessore che esegue le istruzioni del sistema operativo, le funzioni di routing e il controllo delle interfacce; i router più complessi possono avere più CPU;
- RAM: usata per contenere la configurazione corrente del router (*running-config*), la routing table e la cache fast switching, oltre che a svolgere le normali funzioni di memoria volatile; generalmente è di tipo DRAM espandibile con moduli DIMM;
- Flash: viene usata per memorizzare un’intera immagine del Cisco IOS e può essere espansa mediante normali SIMM o schede PCMCIA;
- Bus: nella maggior parte dei router sono presenti due bus, il primo, di *sistema*, col quale la CPU comunica con le interfacce, il secondo, il *CPU bus*, utilizzato per accedere alle memorie;
- NVRAM: contiene la configurazione che viene caricata all’avvio del router (*startupconfig*); non viene persa in caso di spegnimento del router;
- ROM: è la memoria dove è presente il software di base del router e il software di diagnostica: infatti il compito principe della ROM è quello di effettuare l’hardware diagnostic durante il bootup e di caricare l’IOS da flash a RAM;
- Porte: oltre agli slot di espansione prima elencati è presente una porta AUX/Console: sono usate primariamente per la configurazione del router e infatti non sono porte di rete;
- Power supply: fornisce l’energia necessaria per far lavorare le componenti interne e i router di grandi dimensioni possono avere più power supply.

Le immagini seguenti sono tratte da *Cisco Press: CCNA Instructor's Manual* e sono relative a un router della famiglia 2600.

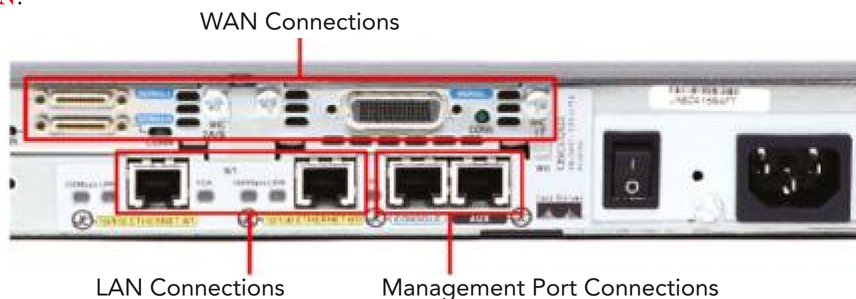


In figura è mostrato l'interno di un ruoter con in evidenza alcuni componenti appena descritti.



La figura evidenzia le interfacce dei router.

L'immagine seguente evidenzia il retro di un router dove sono presenti le interfacce sia di tipo WAN sia di tipo LAN.



La differenza sostanziale tra WAN e LAN consiste nei diversi protocolli utilizzati nello strato fisico e nello strato di data link.

Le interfacce LAN permettono al router di collegarsi solitamente mediante Ethernet oppure il Token Ring o l'ATM.
Le connessioni WAN forniscono connettività attraverso un service provider a un sito distante o a Internet e possono essere di diverso tipo poiché possono usare tutte le diverse tecnologie disponibili.



Zoom su...

CONNESSIONE A WAN

L'apparecchiatura dell'utente, che viene chiamata CPE (Customer Premises Equipment), in genere è un router e svolge la funzione di DTE (Data Terminal Equipment) mediante una connessione al service provider con l'utilizzo di DCE (Data-Circuit terminating Equipment device), solitamente un modem o un CSU/DSU.
Tale device serve a convertire i dati dal DTE dell'utente in una forma accettabile per il WAN service provider, e generalmente il colloquio avviene utilizzando l'interfaccia seriale.

L'ultimo gruppo di porte, quelle indicate come Management Port, comprende la porta console e la porta AUX, e sono porte seriali per la gestione di tipo asincrono:

- la porta console viene usata per visualizzare lo startup del router, il debugging e i messaggi di errore: può servire inoltre per eseguire le procedure di recovery in caso di disastri o per la password recovery. La connessione a tale porta può essere fatta con un comune cavo rollover e un adattatore da RJ-45 a DB-9, e mandando in esecuzione sul PC un programma di emulazione terminale, come Hyperterminal;
- la porta AUX non è presente in tutti i router e viene utilizzata per connettere il router alle reti per compiere monitoraggio o troubleshooting.

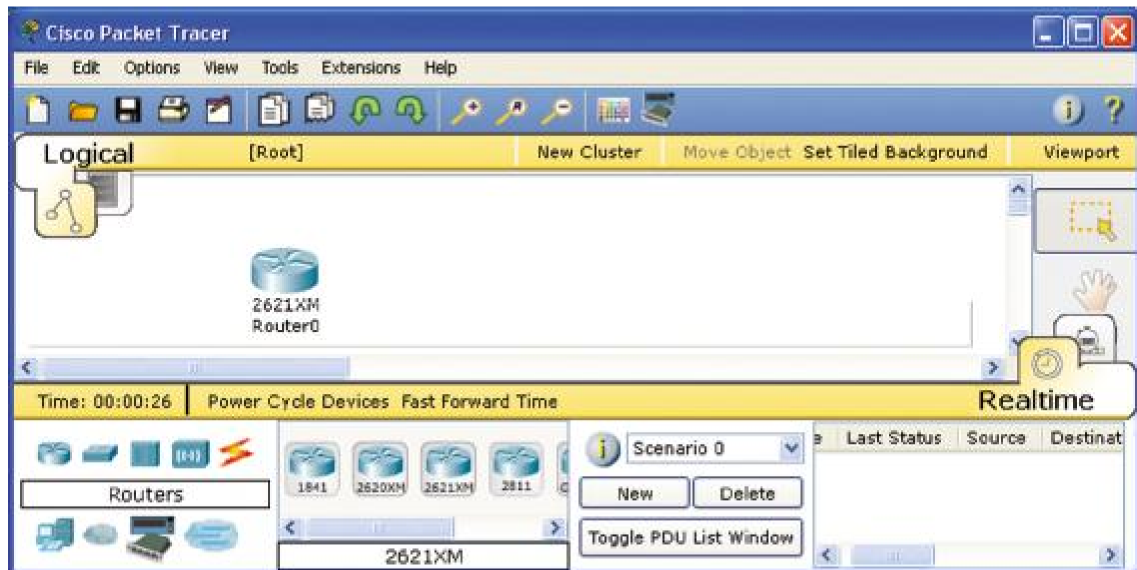
Packet Tracer e l'installazione moduli ai Cisco 2621XM

Tra i router presenti nel simulatore Packet Tracer troviamo due elementi della famiglia 2600: il 2620XM e il 2621XM.

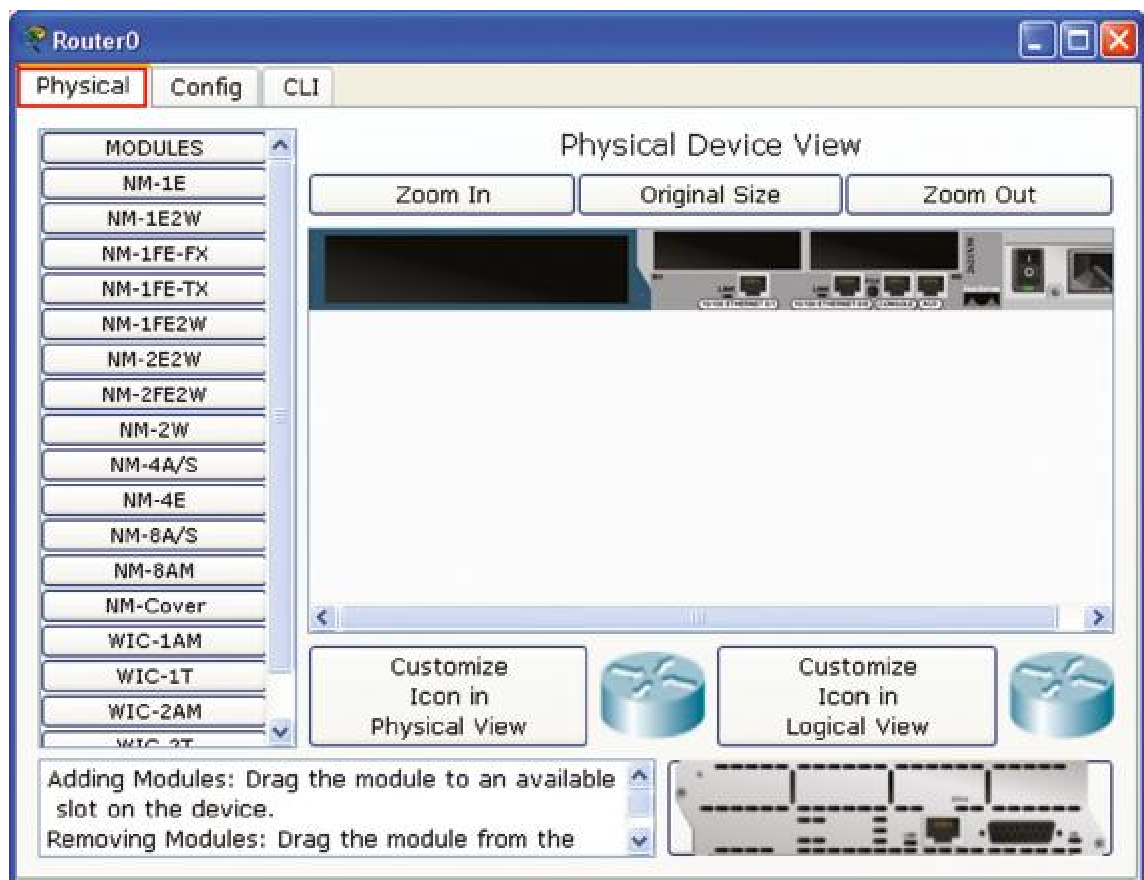


La differenza tra i due modelli consiste semplicemente nel numero di porte Ethernet presenti, che sono una sola nel 2620XM e due nel 2621XM.

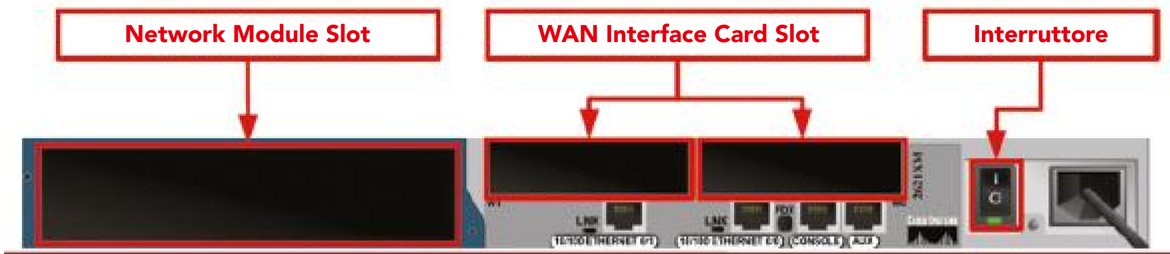
A questi modelli il simulatore Packet Tracer permette di installare moduli aggiuntivi: si seleziona e si trascina il router desiderato nell'area di lavoro, per esempio il 2621XM:



e quindi, cliccandolo con il mouse, si apre la scheda di configurazione dove si sceglie la tab "Physical"



la “vista posteriore” del router mostra tre alloggiamenti:



Per poter aggiungere una scheda di espansione si compiono le seguenti operazioni:
- la prima operazione è quella di spegnere il router, come se fosse un router reale;
- quindi si individua il modulo aggiuntivo che si vuole inserire verificando che ci sia almeno uno slot libero in grado di alloggiarlo;
- infine si trascina il modulo hardware sullo slot libero e si riaccende il router con l’apposito interruttore.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Non è vero che il modello TCP-IP:

a) deriva dal modello OSI
b) è strutturato in 4 livelli
c) è strutturato in 7 livelli
d) è utilizzato in Internet

2 Con internetworking si intende:

a) la connessione tra reti LAN
b) la connessione tra reti WAN
c) la connessione tra reti GAN
d) nessuna delle precedenti

3 Il router svolge le seguenti funzioni fondamentali:

a) creare le tabelle di routing
b) gestire le tabelle di routing
c) risolve gli indirizzi IP
d) determinare il percorso sul quale istradare un pacchetto

4 Con IOS si intende:

a) Internet Operating System
b) Internetwork Operating System
c) Internet Open System
d) Internetwork Open System

5 Quale tra i seguenti acronimi è errato?

a) CPE: Customer Premises Equipment
b) DRE: Data Routing Equipment
c) DTE: Data Terminal Equipment
d) DCE: Data-Circuit terminating Equipment device

6 Quali tra le seguenti tabelle non sono presenti nell'emulatore Packet Tracer?

a) Physical
b) Logical
c) Config
d) Interface
e) CLI

Test vero/falso

1 Una rete di computer è un insieme eterogeneo di dispositivi. V F
2 I protocolli di rete hanno la caratteristica di avere una struttura a livelli sovrapposti. V F
3 La segmentazione logica delle reti viene realizzata mediante i router. V F
4 In un router è sempre presente almeno una porta di ingresso e una porta d'uscita. V F
5 Nel WAN Interface Card Slot vengono aggiunte le porte Ethernet. V F
6 La memoria flash in un router può essere espansa mediante normali SIMM o schede PCMCIA. V F
7 La porta console viene programmata con un programma di emulazione terminale. V F
8 Nel modello 2620XM sono presenti due porte Ethernet. V F

LEZIONE 2

CONFIGURARE I ROUTER

IN QUESTA LEZIONE IMPAREREMO...

- le caratteristiche di un SO per i router
- le modalità operative di IOS
- a utilizzare l'interfaccia CLI di IOS

Il sistema operativo Cisco IOS

Un router è sostanzialmente un personal computer e quindi come ogni personal computer ha bisogno di un sistema operativo per gestire le proprie risorse: tutti i router Cisco utilizzano lo stesso sistema operativo, il Cisco IOS (utilizzato anche negli switch Catalyst).

I servizi offerti dall'IOS Cisco sono i seguenti:
- funzioni base di routing e switching;
- accesso sicuro e affidabile alle risorse della rete;
- scalabilità della rete.

Il sistema IOS è in costante evoluzione e in costante miglioramento e le nuove release introducono migliorie o nuove funzioni: come per ogni PC è quindi necessario aggiornare tale software per poter utilizzare al meglio il proprio apparato.

Tutti gli IOS sono disponibili per il download dal sito Cisco nell'area riservata alla quale si accede dopo il controllo delle credenziali (login e password) che autorizzano il download, dato che l'utilizzo di tale software è subordinato al pagamento di una licenza d'uso.

IOS equipaggia la maggior parte dei router Cisco e anche molti dei suoi switch. Attualmente lo sviluppo di questo software è arrivato alla versione 12.2. Tramite IOS è possibile gestire tutte le caratteristiche del router, dal settaggio degli indirizzi, a quello dei protocolli di routing, al controllo del traffico, all'aggiornamento del software.

A ogni versione di ogni IOS viene dato un nome che segue il seguente schema:

Platform – Features – Run time memory and compression format

con il seguente significato:
- Platform: indica il modello di router per cui il sistema operativo è stato sviluppato;
- Features: sono le caratteristiche e le potenzialità della versione, codificate secondo una tabella specifica;
- Run time memory and compression format: è indicato da due lettere
  – la prima identifica in che area di memoria verrà eseguito il sistema operativo,
  – la seconda il formato di compressione.

ESEMPIO 1

Decodifichiamo la sigla **c7200-ajs56-mz**

Platform

c7200: Router Cisco Serie 7200

Features

a: supporto protocollo APPN;
j: supporto di caratteristiche Enterprise;
s: supporto di NAT,ISL,VPDN/L2F
56: supporto di crittografia a 56 bit

Run time memory and compression format

m: esecuzione in RAM
z: file compresso con Zip



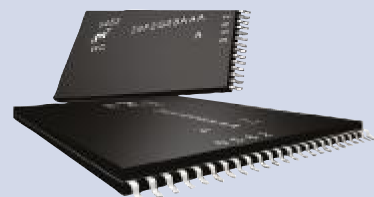
I MARCHI iOS E iPHONE

Dal 2007 Apple vende "iPhone" con sistema operativo "iOS": questi due nomi non li ha inventati Apple, ma sono stati ottenuti grazie a Steve Jobs dalla Cisco. Apple aveva già adottato la "i" prima dei suoi prodotti ma iPhone e iOS erano stati precedentemente registrati da Cisco che però non li stava utilizzando. Adam Lashinsky racconta nel suo libro che Steve Jobs andò da Charles Giancarlo di Cisco e gli disse che voleva il nome iPhone, dato l'inutilizzo del nome da parte di Cisco, e nel 2010 seguì anche la richiesta per iOS che Cisco aveva brevettato come IOS (Internet Operating System). Cisco accettò di concedere in licenza il marchio iOS ad Apple per utilizzare questo nome per il sistema operativo di iPhone, iPod touch e iPad. La licenza è limitata al solo marchio e non comprende alcuna tecnologia, quindi i sistemi operativi non hanno nulla in comune.

Il sistema operativo Cisco IOS viene caricato nel dispositivo copiando la relativa immagine tramite un ◀ **TFTP** ▶ server nella ◀ memoria FLASH ▶ del router: analoga la procedura per l'aggiornamento con le nuove release del sistema.
In molti router, IOS viene copiato nella RAM alla accensione del dispositivo per aumentarne le prestazioni.



◀ Memoria FLASH È un dispositivo di archiviazione di massa utilizzato anche nelle pendrive USB per le sue piccole dimensioni. La memoria di tipo flash è una memoria elettronica, e quindi è più veloce di un normale disco magnetico ed è di tipo non volatile, ovvero mantiene l'informazione anche se non è connessa all'alimentazione elettrica. Un esemplare di memoria flash è riportata a lato. ▶





◀ **TFTP** Trivial File Transfer Protocol (TFTP) is a very simple protocol, simple enough to be implemented in small boot loaders. The basic idea is that when the router gets powered on, then *for a few seconds* it initializes the wired lan ports, and listens to TFTP requests for transferring a flash image. Then either flashes the received image, or continues booting the device normally if the TFTP communication timed out. ▶

Tutti i prodotti della Cisco usano lo stesso IOS ed è possibile controllare quale sia l'immagine di IOS disponibile sulla memoria mediante il comando show version, che mostra un insieme di dettagli tra i quali anche la quantità di memoria RAM che esso occupa:

```
Router>show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang
Image text-base: 0x8000808C, data-base: 0x80A1FECC

ROM: System Bootstrap, Version 12.1(3r)T2, RELEASE SOFTWARE (fc1)
Copyright (c) 2000 by cisco Systems, Inc.
ROM: C2600 Software (C2600-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5)

System returned to ROM by reload
System image file is "flash:C2600-i-mz.122-28.bin"

Cisco 2620 (MPC860) processor (revision 0x200) with 253952K/8192K bytes of memory.

Processor board ID JAD05190MTZ (4292891495)
M860 processor: part number 0, mask 49
Briding software.
X.25 software, Version 3.0.0.
1 FastEthernet/IEEE 802.3 interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)

Configuration register is 0x2102
```

Modalità operative

I device Cisco possono operare in tre modalità:

- ROM monitor;
- Boot ROM;
- Cisco IOS.

Ciascuna ha un proprio ambiente di esecuzione: la modalità di funzionamento viene definita all'atto del caricamento dell'IOS nella memoria RAM dalla procedura di startup, leggendo dal registro di configurazione quello che l'amministratore del sistema ha definito come modalità di default.

Nella modalità ROM monitor viene eseguito il processo di bootstrap che comprende le funzionalità di basso livello e di diagnostica: viene utilizzata per effettuare il recupero in caso di system failures e password perse. A essa si accede esclusivamente tramite console.

Nella modalità Boot ROM sono disponibili solo alcune caratteristiche del Cisco IOS: permette di effettuare le operazioni di scrittura sulla flash memory sostanzialmente solo per sostituire l'immagine dell'IOS, anche utilizzando il comando copy tftp flash.

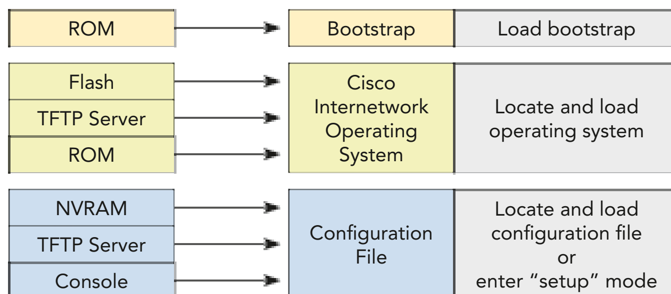
In condizioni normali viene caricato in RAM il sistema operativo memorizzato nella flash e quindi siamo nella terza modalità di funzionamento: Cisco IOS.

Come vedremo in seguito, un comando di sistema visualizza specificamente la quantità di memoria flash utilizzata: show flash.

```
Router>show flash

System flash directory:
File   Lenght    Name/status
3      5571584   c2600-i-mz.122-28.bin
2      28282     sigdef-category.xml
1      227537    sigdef-default.xml
[5827403 bytes used, 58188981 available, 64016384 total]
63488K bytes of processor board System flash (Read/Write)
```

Lo schema seguente, tratto da *Cisco Press: CCNA Instructor's Manual*, offre una panoramica completa delle funzionalità e dei compiti di ogni modalità operativa del router.



Startup del router

Lo startup di un router è simile a quello di un normale computer e avviene caricando il bootstrap, il sistema operativo e il file di configurazione.

All'atto dell'accensione il router compie un power-on self test (POST), esegue i test di diagnostica della ROM e di tutti i moduli hardware, verifica le operazioni base di memoria, la CPU e le interfacce: solo se i test danno esito positivo, il router prosegue con l'inizializzazione del software eseguendo il caricamento del bootstrap, altrimenti segnala la presenza di errori.
Il bootstrap esegue un insieme semplice di istruzioni che testano l'hardware e quindi legge dal campo boot del registro di configurazione la locazione da usare per caricare l'IOS e provvede all'effettivo caricamento di una sua immagine.

Successivamente viene caricato nella memoria principale il file di configurazione letto dalla NVRAM e vengono eseguiti i comandi in esso presenti che avviano il processo di routing, supportano l'indirizzamento per le interfacce e definiscono altre caratteristiche del router.
Se esiste un file di configurazione non valido nella NVRAM allora si cerca nel server TFTP; in caso di ulteriore fallimento si entra nel setup mode.

Setup iniziale

Lo scopo del setup iniziale è esclusivamente quello di permettere all'amministratore di installare una configurazione minimale del router.

Durante la fase di inizializzazione il menu di setup propone le risposte di default in parentesi quadre [ ] seguite dalla domanda, che viene confermata semplicemente premendo Enter, mentre si esce dalla modalità configurazione digitando il comando "Ctrl-C".

Viene richiesta l'immissione delle password per i diversi livelli operativi e proposta la definizione e configurazione delle interfacce.

```
   Enter host name [Router]: router

   The enable secret is a password used to protect access to
   privileged EXEC and configuration modes. This password, after
   entered, becomes encrypted in the configuration.
   Enter enable secret: 1111

   The enable password is used when you do not specify an
   enable secret password, with some older software versions, and
   some boot images.
   Enter enable password: 2222

   The virtual terminal password is used to protect
   access to the router over a network interface.
   Enter virtual terminal password: 3333
 Configure SNMP Network Management? [no]:
 Configuring interface parameters:
 Do you want to configure FastEthernet0/0 interface? [no]:
 Do you want to configure FastEthernet0/1 interface? [no]:
```

Al termine delle operazioni proposte dal processo di configurazione saranno visualizzate le seguenti opzioni:

```
[0] Go to the IOS command prompt without saving this config.
[1] Return back to the setup without saving this config.
[2] Save this configuration to nvram and exit.

Enter your selection [2]:
```



Zoom su...

SCHERMATA DI BOOT

Analizziamo dettagliatamente le informazioni che vengono visualizzate durante il bootup.

```
System Bootstrap, Version 12.1(3r)T2, RELEASE SOFTWARE (fc1)
Copyright (c) 2000 by Cisco Systems, Inc.
PT 1001 (PTSC2005) processor (revision 0x200) with 60416K/5120K bytes of memory
```

Nelle prime tre righe si riconoscono i dati riguardanti la versione dell'IOS, il modello, il tipo di processore, la quantità di memoria.

A queste righe può seguire la seguente indicazione:

NVRAM invalid, possibly due to write erase

la quale indica che il router non è stato ancora configurato; si deve quindi configurare il router per usare il file immagazzinato nella NVRAM.

Le successive informazioni dipendono dalle interfacce del router e dalla release del Cisco IOS

```
Processor board ID PT0123 (0123)
PT2005 processor: part number 0, mask 01
Bridging software.
X.25 software, Version 3.0.0.
4 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
32K bytes of non-volatile configuration memory.
63488K bytes of ATA CompactFlash (Read/Write)
```

Si individua il numero di interfacce, il tipo, la quantità di memoria NVRAM e FLASH.

Command Line Interface CLI

L'IOS Cisco usa una interfaccia a linea di comando (CLI) che permette di accedere direttamente alla funzionalità del sistema operativo, che varia a seconda della versione dell'IOS e del tipo di dispositivo.

Ci sono tre possibili modalità di accesso al CLI.

- Console: noto anche come linea CTY, usa una connessione seriale a bassa velocità presente in ogni computer oppure un accesso remoto mediante l'utilizzo di un modem; non è necessario che il router abbia i servizi di rete configurati e quindi questa modalità viene utilizzata per la messa in rete dei servizi non ancora inizializzati e per la prima messa in funzione del router, oltre alle operazioni di manutenzione per risolvere qualsiasi problema che si verifichi durante il funzionamento.
  Possiamo quindi riassumere le principali operazioni effettuate dalla console:
  - configurazione iniziale del dispositivo di rete;
  - procedure di disaster recovery e risoluzione dei problemi quando non è possibile effettuare l'accesso remoto;
  - procedure di recupero password.

Il dispositivo dovrebbe essere situato in una stanza chiusa o in un rack attrezzato per impedire l'accesso fisico a chiunque non sia autorizzato.

- Telnet e SSH: per stabilire una sessione Telnet verso un router almeno una interfaccia deve essere configurata con un indirizzo Layer 3, come per esempio IPv4, e il virtual terminal deve essere configurato per il login e la password. I dispositivi che utilizzano Cisco IOS includono un processo server Telnet che viene mandato in esecuzione all'avvio del dispositivo e per avviare tale sessione è necessario autenticarsi con password.

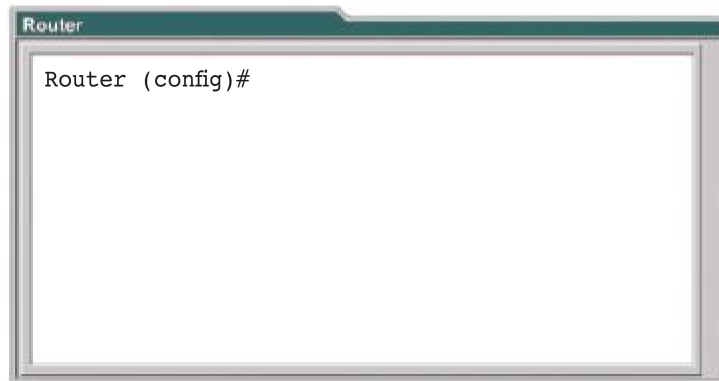
Il protocollo Secure Shell (SSH) è il metodo più sicuro di accesso remoto al dispositivo: prevede la struttura per un login remoto simile a Telnet, tranne per il fatto che utilizza i servizi di rete più sicuri dato che effettua la crittografia di tutte le comunicazioni tra il dispositivo e il client.

È sempre consigliato utilizzare SSH al posto di Telnet dato che nelle maggior parte delle versioni di IOS è presente e in alcuni dispositivi è abilitato di default.

- AUX: un altro modo per stabilire una sessione CLI è quello di effettuarla in remoto, tramite un telefono e un modem con connessione dialup, collegato appunto alla porta AUX del router.
  È molto simile alla connessione console e deve essere disponibile e configurata sul dispositivo: inoltre la porta AUX può anche essere utilizzata in connessione locale, con una connessione diretta a un computer in cui sia in esecuzione un programma di emulazione terminale.

La porta console è sempre da preferirsi alla porta AUX per la configurazione del router e per la risoluzione dei problemi anche perché visualizza automaticamente lo stato del router all'avvio, i messaggi di debugging e di errore; l'unico effettivo utilizzo della porta AUX al posto della porta console è quando quest'ultima presenta problemi di funzionamento oppure quando non sono noti alcuni parametri della porta console stessa.

Ciascuna modalità appena descritta può utilizzare come programma di comunicazione Windows Hyperterminal: nella figura seguente è riportata una tipica finestra di comunicazione.



Modalità di funzionamento dell'IOS

Il sistema operativo Cisco IOS è progettato a moduli, ciascuno con una modalità di accesso e un determinato set di istruzioni e di funzionalità.
Il CLI offre modalità di funzionamento organizzate in una struttura gerarchica e richiede accessi diversi per operare in ciascuna modalità; le principali sono:
- user command EXECutive mode;
- privileged command EXECutive mode (o enable mode);
- global configuration mode;
- altre specifiche modalità di configurazione.

Per garantire la sicurezza è possibile configurare ogni modalità in modo da permettere l'accesso a utenti differenziati in base al loro livello di autenticazione: sono presenti comunque alcune funzionalità comuni disponibili indipendentemente dal livello di sicurezza.

Le sessioni EXEC (command executive), cioè il servizio di interpretazione dei comandi dell'IOS, prendono il nome di modalità primarie e hanno due livelli di accesso:

- user EXEC mode ("view only" mode);
- privileged EXEC mode o enable mode;

che si individuano per la forma del prompt che viene visualizzato sulla linea di comando.

Di default ogni prompt inizia con il nome del dispositivo ed è seguito da un carattere che specifica la modalità di funzionamento:

- la user EXEC si identifica con il prompt ">", permette solo un limitato numero di comandi base per funzioni di monitoraggio e non comprende nessun comando che possa cambiare la configurazione del router;
- il privileged EXEC si riconosce per il prompt "#" e consente l'uso dell'intero insieme di comandi: per accedere è possibile abilitare una password di sicurezza.

> Per passare dal modo user al modo privileged e viceversa si digita al prompt il comando enable e disable rispettivamente (oppure il comando exit, comune per entrambe le modalità).

Router

```
Router con0 is now available

Press RETURN to get started.

User Access Verification
Password:
Router>          <------ User-Mode Prompt
Router>enable
Password:
Router#          <------ Privileged-Mode Prompt
Router#disable
Router>
Router>exit
```

La separazione della modalità di comando in due tipi di accesso viene effettuata per motivi di sicurezza e ogni gruppo ha comandi simili ma la modalità privileged EXEC ha un livello maggiore di autorità intesa come opzioni di livelli di configurazione.

User Executive Mode

La user executive mode, o sinteticamente indicata come user EXEC, si limita alle operazioni basilari con un insieme ridotto di comandi, e si trova al top della struttura gerarchica ed è la modalità di default che viene presentata all'utente al primo accesso al sistema IOS.
Una parte dei comandi disponibili permette solo la visualizzazione dello stato e non offre opzioni di settaggio o di modifica della configurazione del dispositivo.
Per questo motivo, dato il basso livello di operatività, di default non è prevista nessuna password per accedere a questa modalità, ma è consigliato aggiungere un livello di protezione anche per essa durante la fase di configurazione iniziale del router.

Digitando il punto interrogativo (?) nella modalità EXEC si visualizza una lista dei comandi disponibili per la modalità corrente, come nell'immagine seguente che riporta le alternative in modalità user:

```
Router>?
Exec commands:
   <1-99>        Session number to resume
   connect       Open a terminal connection
   disable       Turn off privileged commands
   disconnect    Disconnect an existing network connection
   enable        Turn on privileged commands
   exit          Exit from the EXEC
   logout        Exit from the EXEC
   ping          Send echo messages
   resume        Resume an active network connection
   show          Show running system information
   ssh           Open a secure shell client connection
   telnet        Open a telnet connection
   terminal      Set terminal line parameters
   traceroute    Trace route to destination
```

Privileged EXEC Mode

Per poter effettuare la configurazione e la manutenzione del dispositivo è necessario essere in modalità privileged EXEC: anch'essa non prevede di default nessuna password operativa, ma date le possibilità offerte da questa modalità al primo avvio è necessario aggiungere una password.
Infatti, solo in modalità privileged EXEC è possibile effettuare la configurazione generale del router e il settaggio dei componenti specifici e delle interfacce e quindi è necessario proteggere tale accesso per evitare modifiche indesiderate sia per errori operativi che per accesso da parte di non autorizzati.
Analogamente alla modalità precedente, digitando ? si visualizza una lista dei comandi disponibili:

```
Router>enable
Router#?
Exec commands
   <1-99>        Session number to resume
   auto          Exec level Automation
   clear         Reset functions
   clock         Manage the system clock
   configure     Enter configuration mode
   connect       Open a terminal connection
   copy          Copy from one file to another
   debug         Debugging functions (see also 'undebug')
   delete        Delete a file
   dir           List files on a filesystem
   disable       Turn off privileged commands
   disconnect    Disconnect an existing network connection
   enable        Turn on privileged commands
   erase         Erase a filesystem
   exit          Exit from the EXEC
   logout        Exit from the EXEC
   mkdir         Create new directory
   more          Display the contents of a file
```

```
   no           Disable debugging informations
   ping         Send echo messages
   reload       Halt and perform a cold restart
   resume       Resume an active network connection
   rmdir        Remove existing directory
   setup        Run the SETUP command facility
   show         Show running system information
   ssh          Open a secure shell client connection
   telnet       Open a telnet connection
   terminal     Set terminal line parameters
   traceroute   Trace route to destination
   undebug      Disable debugging functions (see also 'debug')
   write        Write running configuration to memory, network, or terminal
Router#
```

Per passare in modo privilegiato basta anche digitare sinteticamente "ena" al prompt in modalità user.

Per ripetere l'ultimo comando si preme "Ctrl-P" o "freccia in su": un errore di digitazione o un comando non riconosciuto viene evidenziato con il simbolo ^.

I comandi che vengono inseriti in modalità EXEC hanno una struttura gerarchica: al comando di primo livello segue l'insieme delle opzioni disponibili di secondo e di terzo livello, come riportato in figura.

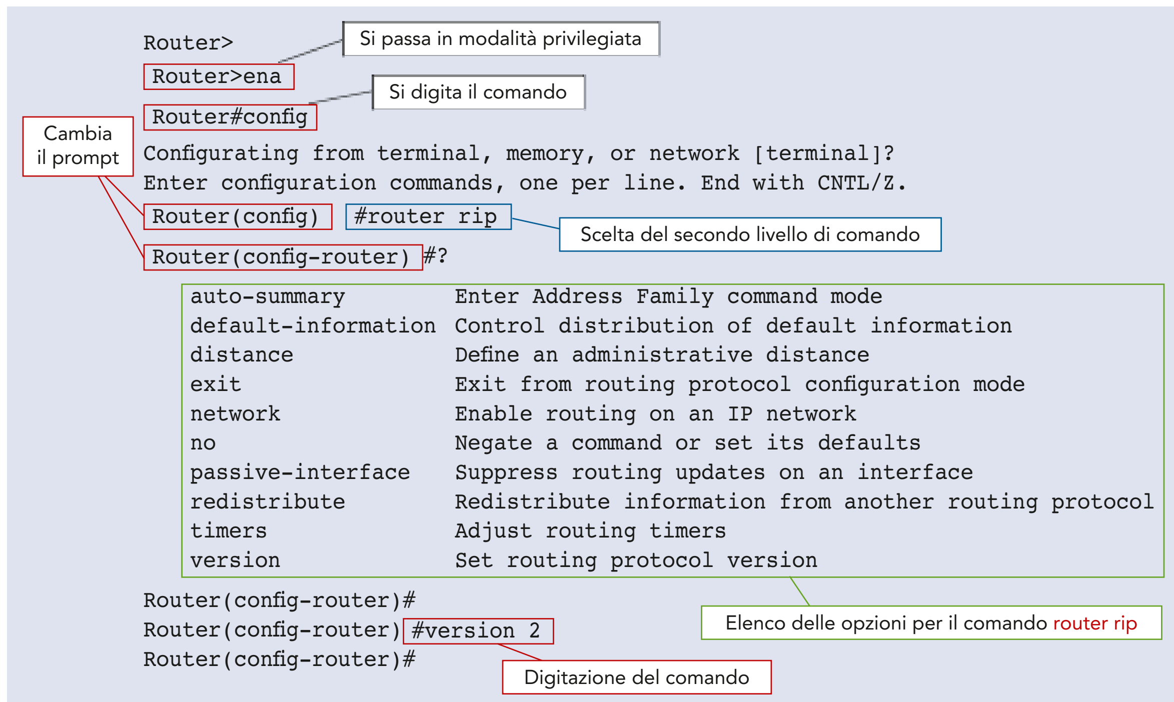
```
Privileged EXEC Commands-Router#
all User EXEC Commands
debug commands
reload
configure
etc…
        Global Configuration Commands-Router(config)#
        hostname
        enable secret
        ip route
        interface   ethernet
                    serial
                    bri
                    etc.
                            Interface Commands-Router(config-if)#
                            ip address
                            ipx address1
                            encapsulation
        router      rip
                    ospf
                    eig rp
                    etc.
                            Routing Engine Commands-Router(config-router)#
                            network
                            version
                            auto summary
                            etc…
        line        vty
                    console
                    etc.
                            Line Commands-Router(config-line)#
                            password
                            login
                            modem commands
                            etc…
```

A ogni livello di comando viene anche modificato il prompt in modo che l'utente abbia una visione della posizione gerarchica in cui il sistema si trova e sia facilitato nella scelta dei comandi da inserire.

Vediamo un esempio di comando privilegiato, indicando per ogni opzione l'albero delle sue alternative.

ESEMPIO 2

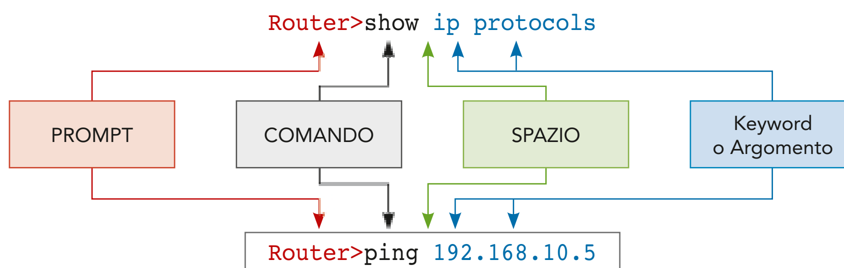
Settiamo a **2** la versione del protocollo RIP.



Esempi di comandi CLI

Riportiamo alcuni esempi di comandi CLI, alcuni dei quali saranno successivamente utilizzati nelle lezioni di laboratorio.

Abbiamo detto che ogni comando IOS ha un formato e una sintassi specifica a seconda della modalità di esecuzione: la struttura generale è quella formata dallo schema seguente. ▼



I comandi non sono case-sensitive quindi non fanno differenza tra lettere maiuscole e minuscole e generalmente ammettono delle abbreviazioni.

Abbiamo visto il passaggio da modo User a Privileged con il comando enable, che può essere scritto nella forma sintetica:

```
Router>ena
```

Analogamente il comando che visualizza le statistiche sulle interfacce presenti può essere scritto

```
Router>show interfaces
```

o nella forma sintetica

```
Router>show int
```

Se al comando viene aggiunto un secondo argomento viene aumentata la granularità delle informazioni visualizzando solo il dettaglio della opzione richiesta; per esempio volendo visualizzare solo le informazioni dell'interfaccia seriale 0/1 il comando completo è il seguente:

```
Router>show int serial 0/1
```

Vediamo ora come assegnare al router un nome, che deve essere univoco nella LAN:
come primo passo entriamo in modalità Privileged

```
Router>ena
```

quindi entriamo nel sottomenu di configurazione

```
Router#config
```

e infine assegniamo il nome desiderato

```
Router(config)#hostname Napoli
```

Da ora in poi il prompt assume il nome del router, cioè si presenta nel seguente formato:

```
Napoli#
```

Aggiungiamo ora una password alla modalità virtual terminal line:

```
Napoli#config
Napoli(config)#
Napoli(config)#line vty 0 4
Napoli(config-line)#password alibaba
```

Nel prossimo esempio configuriamo una interfaccia seriale dove è necessario settare il clock per la sincronizzazione nelle comunicazioni con il comando "clock rate <rate[bit/s]>":

```
Napoli#config
Napoli(config)#
Napoli(config)#interface serial 0/1
Napoli(config-if)#clock rate 56000
Napoli(config-if)#no shutdown
```

L'ultimo comando è quello che rende attiva l'interfaccia sul router.

Configuriamo ora una interfaccia ethernet, dove non è necessario configurare il clock rate dato che è una interfaccia asincrona:

```
Napoli#config
Napoli(config)#
Napoli(config)#interface e0
Napoli(config-if)#ip address 180.6.130.2 255.255.255.128
Napoli(config-if)#no shutdown
```

Come ultimo esempio associamo un host name di un dispositivo presente sulla rete a un indirizzo IP:

```
Napoli#config
Napoli(config)#ip host Vesuvio 180.6.130.7
Napoli(config)#ip host Portici 90.6.0.17
Napoli(config)#ip host Maradona 170.6.4.2
```

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 La sigla c7200-ajs56-mz indica:

a) la versione di IOS per gli switch Catalyst 720.0
b) che il file è compresso con arj
c) che deve essere caricato in memoria RAM
d) che occupa 56 kb di RAM

2 TFTP significa:

a) Transfer Flash Trivial Protocol
b) Transfer File Trivial Protocol
c) Trivial Flash Transfer Protocol
d) Trivial File Transfer Protocol

3 I device Cisco possono operare in tre modalità (indicare quella errata):

a) ROM monitor
b) Cisco IOS
c) Boot ROM
d) ROM IOS

4 Ordina le operazioni di startup di un router:

a) .... eseguendo il caricamento del bootstrap
b) .... esegue i test di diagnostica dalla ROM
c) .... verifica le operazioni base di memoria, la CPU e le interfacce
d) .... esegue i test di diagnostica dei moduli hardware
e) .... esegue POST

5 Nelle prime tre righe della schermata di boot si riconoscono (più risposte corrette):

a) versione dell'IOS
b) il numero di porte
c) il modello
d) l'indirizzo IP
e) il tipo di processore
f) la quantità di memoria
g) l'indirizzo delle interfacce

6 L'acronimo LCI sta a indicare:

a) Command Language Interface
b) Command Line Instruction
c) Command Language Interface
d) Command Line Interface
e) Command Language Instruction

Test vero/falso

1 Il software IOS viene rilasciato con licenza d'uso open source. V F
2 Platform indica il modello di router per cui il sistema operativo è stato sviluppato. V F
3 Il sistema IOS è memorizzato nella memoria FLASH del router. V F
4 Il sistema iOS di Apple deriva da IOS di Cisco. V F
5 La modalità di funzionamento viene letta al boot dal registro di configurazione. V F
6 Allo starter il menu di setup propone le risposte di default in parentesi quadre [ ]. V F
7 Il modo console può essere eseguito mediante accesso remoto. V F
8 La porta console è sempre da preferirsi alla porta AUX per la configurazione del router. V F
9 La user EXEC si identifica con il prompt "#". V F
10 Le sessioni EXEC o command executive prendono il nome di modalità secondarie di funzionamento. V F

Esercitazioni di laboratorio 1

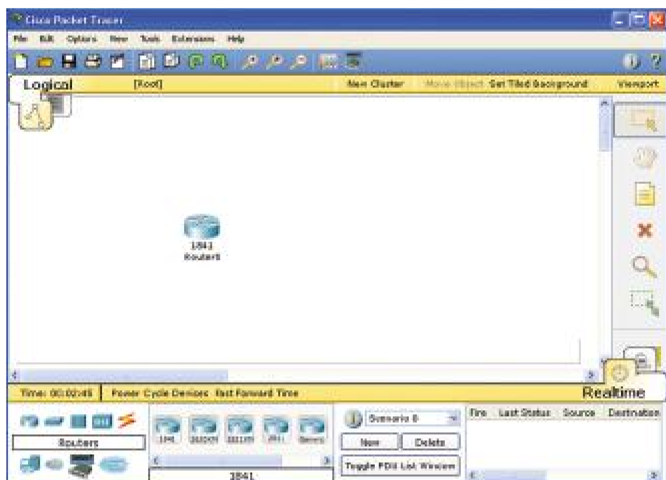
I ROUTER CON PACKET TRACER

In queste esercitazioni utilizzeremo il software di simulazione Packet Tracer per realizzare alcune reti con router, in modo da prendere familiarità con i comandi che permettono di configurare situazioni anche complesse.

> Packet Tracer (PT) è uno software didattico distribuito liberamente agli studenti e agli istruttori del programma Cisco Networking Academy, scaricabile al sito http://cisco.netacad.net, dietro semplice registrazione presso una Academy. È stato descritto nell'esercitazione di laboratorio 4, Modulo 6, del Volume 1 di Sistemi e Reti. L'esercitazione è anche disponibile in rete, all'indirizzo www.hoepliscuola.it, nella sezione riservata al presente volume.

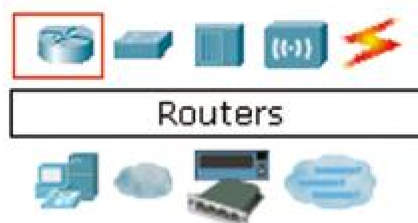
Il software PT consente di:
- creare delle topologie composte da dispositivi generici e/o Cisco;
- simulare la CLI Command Line Interface del sistema operativo Cisco IOS;
- svolgere delle analisi di tipo "what if" creando scenari di traffico e osservando il corrispondente comportamento della rete;
- ispezionare dinamicamente in ogni momento lo stato di ciascun dispositivo e il formato di ciascun pacchetto attivo sulla topologia di rete.

ESERCIZIO 1: configurazione di un router



Manda in esecuzione PT e inserisci un router nell'area di lavoro, come nella figura a lato: ◀

Seleziona dapprima l'icona del router tra i dispositivi nella sezione in basso a sinistra della videata. ▼



Trascina il router 1841 all'interno dell'area di lavoro ▶

1841 2620XM 2621XM 2811 Generic

1841

Posizionando il mouse sopra il router viene visualizzata la scheda con la configurazione corrente del dispositivo: ▼

1841
Router0

Port	Link	VLAN	IP Address	IPv6 Address	MAC Address
FastEthernet0/0	Down	--	<not set>	<not set>	0003.E4A6.E701
FastEthernet0/1	Down	--	<not set>	<not set>	0003.E4A6.E702
Vlan1	Down	1	<not set>	<not set>	00D0.97E6.4D2C

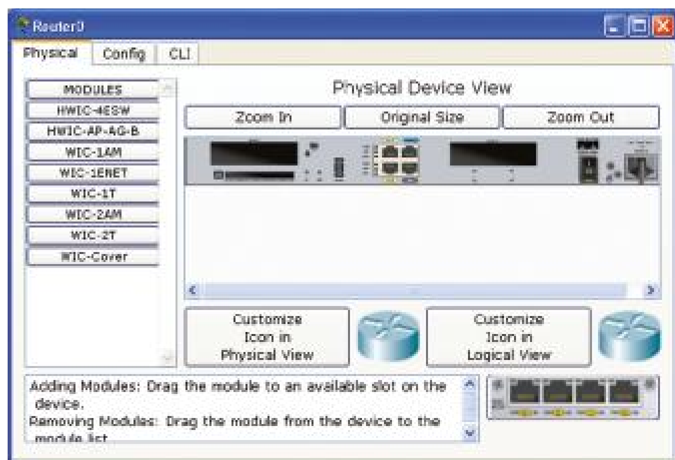
Hostname: Router

Physical Location: Intercity, Home City, Corporate Office, Main Wiring Closet

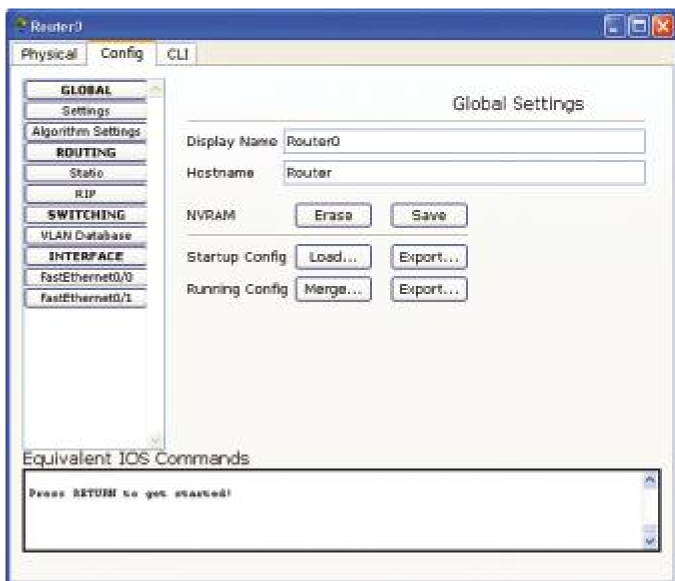
che in questo momento risulta essere non ancora definita.

Cliccando su di esso si apre una scheda composta da tre sezioni.

- Phisical: permette di modificare l'hardware del dispositivo aggiungendo o eliminando moduli come schede di rete, interfacce ecc. presenti nell'elenco di sinistra, semplicemente trascinandoli nell'area di lavoro. ▶ In questa scheda è anche possibile accendere o spegnere il dispositivo.



- Config: permette di effettuare la configurazione di base del dispositivo. ▶ Sono presenti quattro sezioni :
  – GLOBAL: permette di dare un nome al router (noi gli daremo *Volume2*) e di salvare tutte le impostazioni nella memoria NVRAM, farne una copia di backup (genera un file dal nome *Volume2_startup-config.txt*) ed effettuarne successivamente il restore;
  – ROUTING: in questa sezione si impostano le rotte statiche e i protocolli RIP;
  – SWITCHING: come vedremo in seguito, in questa scheda si può configurare una VLAN;
  – INTERFACE: in questa sezione si configurano le interfacce.



Ogni operazione può essere fatta anche manualmente digitando il comando IOS che viene visualizzato nella sezione inferiore, l'Equivalent IOS Commands.

- CLI (Command Line Interface): quest'ultima sezione simula la linea di commando del sistema operativo IOS degli apparati CISCO.

ESERCIZIO 2: assegnazione di un nome al router

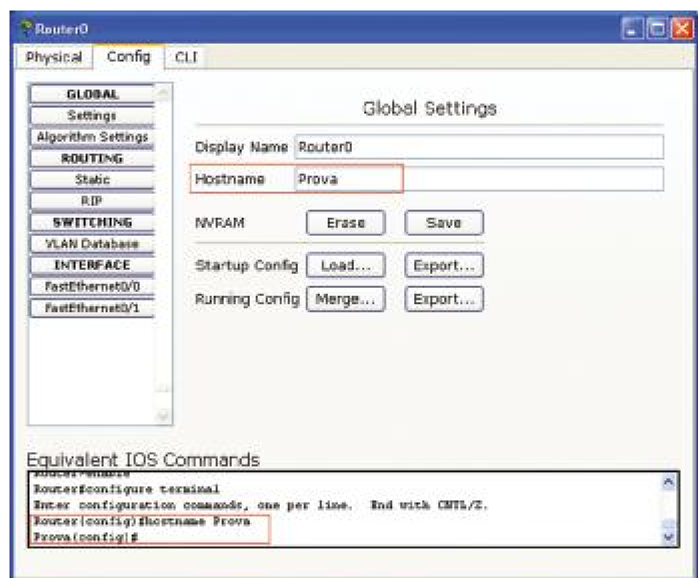
Per assegnare un nome al router abbiamo due possibilità:

A utilizzare la scheda Config di configurazione assistita;

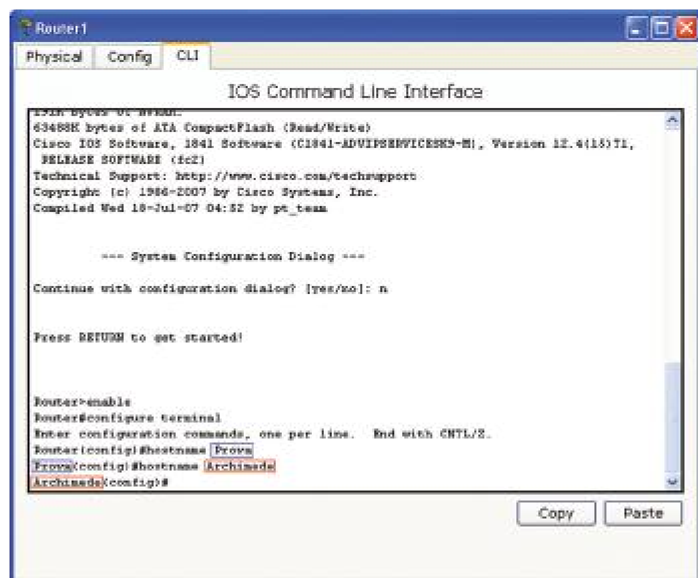
B digitare i comandi nella scheda CLI.

Vediamo entrambe le alternative, iniziando con lo scrivere il nome desiderato (nel nostro esempio Prova) nella scheda Config ▶

Nella finestra inferiore, la Equivalent IOS Commands, viene visualizzata la sequenza di comandi IOS che produrrebbero la medesima operazione se digitati nella CLI.



Cambiamo ora il nome al router scrivendo proprio gli stessi comandi nella scheda CLI: ▶



ESERCIZIO 3: settaggio delle password

Come successiva operazione limitiamo l'accesso al router effettuando la configurazione di password che possono essere messe:

1. al router per accedere dalla porta console;
2. al router da una virtual terminal line (vty), per l'accesso tramite Telnet;
3. alla modalità Privileged EXEC.

1 Per configurare una password per l'accesso tramite console:

```
Archimede(config)#
 Archimede(config)#line console 0
 Archimede(config-line)#password Pitagorico
Archimede(config-line)#
```

2 Per configurare una password per l'accesso tramite telnet (virtual terminal line):

```
Archimede(config-line)#
 Archimede(config)#line vty 0 4
 Archimede(config-line)#password Siracusa
Archimede(config-line)#
```

3 Per configurare una password per l'accesso tramite console modalità Privileged EXEC sono possibili due soluzioni:

A Attraverso il comando "enable password"

```
Archimede(config)#
 Archimede(config)#enable password Euclide
Archimede(config)#
```

In questo caso la password è memorizzata "in chiaro" (non cifrata) nel file di configurazione (è visibile attraverso "show running-config" o "show startup-config")

```
 Archimede#show running-config
Building configuration…

Current configuration : 543 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Archimede
!
!
!
 enable password Euclide
```

È possibile cifrare successivamente la password che abbiamo settato abilitando il servizio di cifratura con il seguente comando:

```
Archimede(config)#service password-encryption
```

B Attraverso il comando "enable secret"

```
Archimede(config)#
Archimede(config)# enable secret Erone
Archimede(config)#
```

La password viene cifrata nel file di configurazione: se infatti visualizziamo ora la configurazione non sarà visibile la password in chiaro:

```
Password:
Archimede# show running-config
Building configuration…

Current configuration : 590 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Archimede
!
!
!
enable secret 5 $1$mERr$tMhlatjzsOH8KastaCQyEl
```

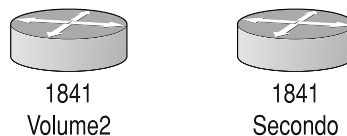
Esercitazioni di laboratorio 2

CONNESSIONE DI DUE ROUTER

In questa esercitazione vedremo come è possibile connettere tra loro due router mediante dapprima un collegamento seriale e quindi tramite una linea ethernet.

Collegamento seriale

Inseriamo nell'esempio precedente un secondo router Cisco1841 e lo chiamiamo Secondo.

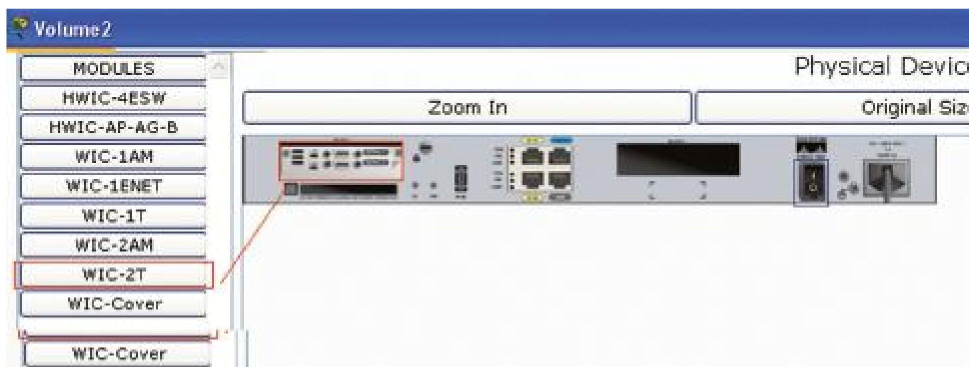


L'assegnazione del nome la effettuiamo utilizzando la linea di comando del CLI:

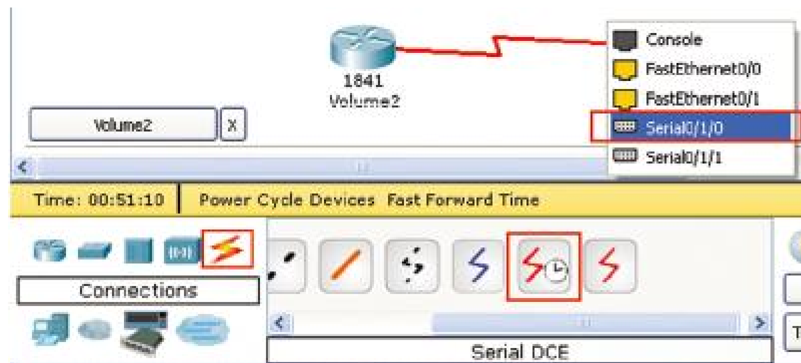
```
Router>ena
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Secondo
Secondo(config)#
```

Aggiungiamo a ogni router una scheda con l'interfaccia seriale, per esempio la scheda W1C-2T che ne possiede 2, trascinandola su di uno slot libero.

Per poter inserire la scheda è necessario dapprima spegnere il dispositivo, posizionando l'interruttore su OFF.



Ora riaccendiamo i router e colleghiamoli con un cavo seriale DCE selezionando dapprima il gruppo Connections nel primo menu e quindi il cavo indicato in figura.
Cliccando sopra i router vengono elencate le porte disponibili: effettuiamo la connessione alla porta Serial 0/1/0 in entrambi.



Se l'operazione è andata a buon fine i due router saranno connessi come nell'immagine a fianco ▶, con acceso il "pallino verde" che simula il led di stato di livello 1.

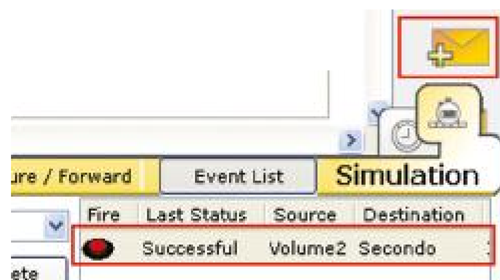


Lo stesso risultato si ottiene scegliendo dal secondo menu il "collegamento automatico" selezionando l'icona seguente: ▶



Verifichiamo ora la connessione, con un semplice ping, dopo aver assegnato due indirizzi IP alle schede, ad esempio *192.168.1.1* e *192.168.1.2*.

Cliccando prima sulla busta del menu laterale destro e successivamente sui due router



inizia la simulazione e appare l'animazione come catturata in figura. ▶



Tutti i file con gli esempi proposti sono disponibili sul sito nella sezione relativa al presente volume: questo esempio è salvato nel file Router1.pkt

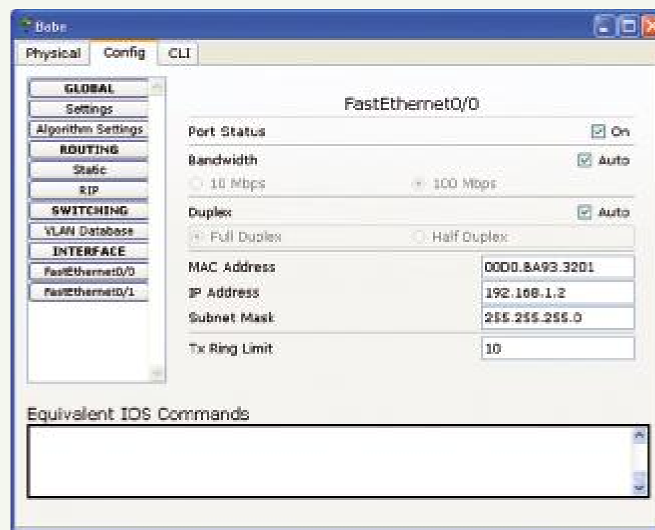
Collegamento ethernet



Prova adesso!

Crea ora una nuova situazione con due nuovi router e assegna loro i nomi Ali e Baba: successivamente effettua tra di essi la connessione sulla porta ethernet 0, dopo averla opportunamente configurata:

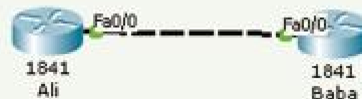
- nel router Ali configurala nella scheda Config;



- nel router Baba configurala con la linea di comando in questo modo:

```
Router#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#
```

Quindi effettua il collegamento tra i due router selezionando il cavo opportuno; quando i due led sono verdi si è pronti per iniziare la simulazione per verificarne la connessione:



Avvia la simulazione sempre mediante un comando PING che produrrà la seguente situazione:



La soluzione la puoi trovare nel file Router2.pkt

2 IL ROUTING: PROTOCOLLI E ALGORITMI

UNITÀ DI APPRENDIMENTO

OBIETTIVI

- Conoscere le problematiche connesse all'instradamento
- Conoscere il concetto di instradamento diretto e indiretto
- Saper interpretare una tabella di routing
- Conoscere la differenza tra routing statico e dinamico
- Conoscere le tipologie degli algoritmi statici
- Comprendere il concetto di Autonomous System (AS) e routing gerarchico
- Apprendere i protocolli IGP: RIP e OSPF
- Apprendere un protocollo EGP: il BGP

ATTIVITÀ

- Configurare manualmente una tabella di routing
- Individuare l'analogia tra reti e grafi
- Saper effettuare la ricerca del cammino minimo (shortest path)
- Individuare la relazione tra grafi, alberi e spanning tree ottimo
- Saper applicare le politiche di instradamento
- Applicare l'algoritmo di Dijkstra
- Applicare l'algoritmo di Bellman-Ford

Lezione 1

FONDAMENTI DI ROUTING

IN QUESTA LEZIONE IMPAREREMO...

- il concetto di instradamento diretto e indiretto
- a costruire le tabelle di routing

Introduzione

Il livello network (o livello 3) della pila ethernet è incaricato di muovere i pacchetti dalla sorgente attraversando tanti sistemi intermedi della subnet di comunicazione fino a che il messaggio inviato dal mittente giunge alla destinazione finale.

Le operazioni effettuate a questo livello sono molto più complesse di quelle svolte dal livello data link, i quali compiti sono esclusivamente quelli di "spostare informazioni" solo da un capo all'altro di una connessione (o segmento di rete): i pacchetti nei quali viene scomposto il messaggio possono raggiungere host anche non presenti sulla rete locale ma appartenenti ad altre LAN remote: devono quindi attraversare una o più MAN.

Il livello network, in sintesi, deve:
- conoscere la topologia della rete;
- scegliere di volta in volta il cammino migliore (instradamento o routing);
- gestire il flusso dei dati (flow control);
- gestire le congestioni (congestion control);
- gestire le problematiche dovute alla presenza di più reti (internetworking).

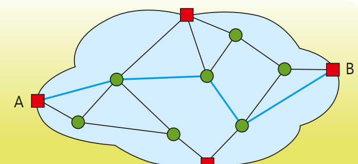
Il routing: concetti generali

L'instradamento (o routing) è alla base della funzionalità di rete implementata dalle entità di livello 3.



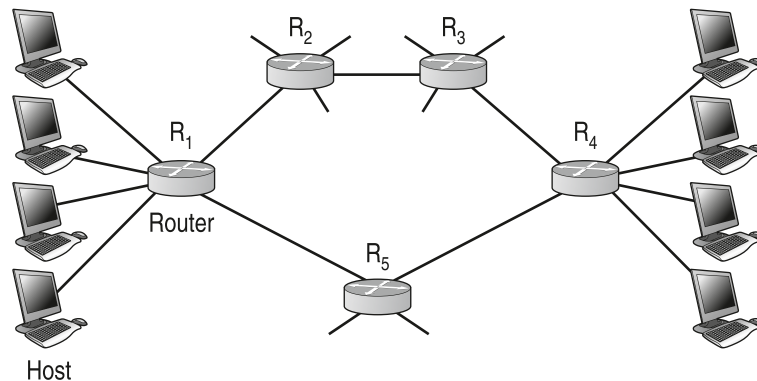
ROUTING

Il routing consente a due nodi A e B, non collegati direttamente, di comunicare tra loro mediante la collaborazione di altri nodi posti su un cammino nella rete che connette A e B.



Le entità di livello 3 effettuano la commutazione (forwarding) verso il SAP (Service Access Point) d'uscita sulla base di un indirizzo (o di una etichetta) posta sul pacchetto e analizzando una tabella presente nella loro memoria, la tabella di instradamento (o ◀ **routing** ▶ table), che contiene le informazioni sul "cammino" che deve prendere il pacchetto.

Possiamo quindi distinguere i dispositivi in due tipologie:
1. i terminali della rete, che sono denominati Host;
2. i nodi di commutazione, che sono denominati Router (anche chiamati IS Intermediate System).

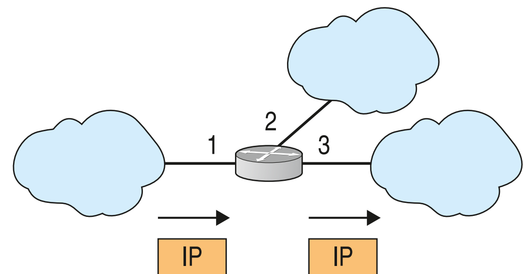




◀ **Routing** Is the process of moving a packet of data from source to destination; is usually performed by a dedicated device called a router. ▶

Tipi di instradamento

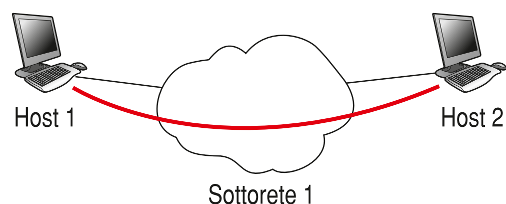
I router sono dei dispositivi di internetworking con interfacce di uscita multiple: hanno la funzione di instradare i datagrammi in rete adottando la tecnica del *routing by network address*: ricevono un datagramma da una interfaccia che contiene l'indirizzo univoco di un sistema presente sulla rete, lo analizzano e lo "rispediscono" sulla rete mediante un'altra interfaccia. Ogni datagramma attraversa un cammino composto da router e sottoreti.



L'instradamento può essere di due tipi:
- instradamento diretto (o routing implicito): hanno di solito più di una interfaccia dove poter effettuare l'inoltro diretto;
- instradamento indiretto (o routing esplicito): si basa su *tabelle di routing* dove è definita la "rotta" di instradamento.

Instradamento diretto (o forwarding diretto)

Siamo nella situazione di instradamento diretto quando la trasmissione di un datagramma IP avviene tra due host connessi su una singola rete logica IP (stesso netID) e quindi **non** coinvolge router intermedi. Il trasmettitore IP risolve l'indirizzo fisico MAC dell'host destinatario tramite il protocollo ARP, incapsula il datagramma nell'unità dati della rete fisica e lo invia verso destinazione.



Prende anche il nome di destination based routers: l'inoltro IP è basato sul solo indirizzo di destinazione.

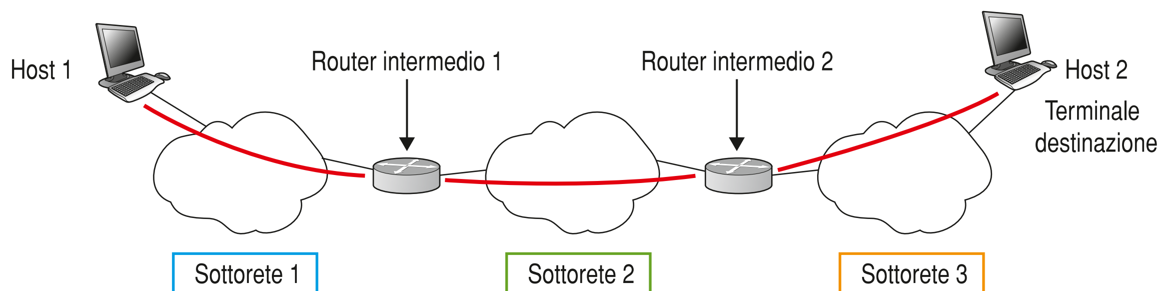
Instradamento indiretto (o forwarding indiretto)

Siamo in presenza dell'instradamento indiretto quando l'host di destinazione non è connesso direttamente alla rete alla quale appartiene l'host mittente e quindi l'unica possibilità per raggiungere la destinazione è quella di passare attraverso uno o più router: il mittente deve fornire però l'indirizzo del prossimo router (◀ **next hop** ▶), oppure l'indirizzo di default gateway, e quando questo riceve il pacchetto IP, lo esamina e decide a quale altro router deve essere inviato; questo procedimento si ripete nei successivi router fino a che il pacchetto giunge alla sottorete di destinazione.



◀ **Next hop** A next hop is the next router to which a packet is sent from any given router as it traverses a network on its journey to its final destination. ▶

Nella sottorete di **destinazione** è utilizzato l'instradamento diretto **e nelle tabelle di routing è indicato solo il prossimo router** (next hop) **nel percorso verso la destinazione** (◀ Next hop routing ▶).





Zoom su...

INDIVIDUAZIONE DEL TIPO DI INDIRIZZAMENTO

Per determinare se l'instradamento è diretto o indiretto un host ispeziona semplicemente il netID effettivo applicando la maschera di sottorete all'indirizzo IP di destinazione.

- Corrispondenza: se c'è corrispondenza con l'indirizzo della sottorete alla quale anch'esso appartiene avviene l'instradamento diretto e l'host può inoltrare il pacchetto direttamente senza passare attraverso router risolvendo l'indirizzo IP in un indirizzo ethernet con l'ARP (Address Resolution Protocol).
- Non corrispondenza: in questo caso siamo in presenza dell'instradamento **indiretto** e l'host "avvia" il pacchetto al gateway di default verso il router più vicino.

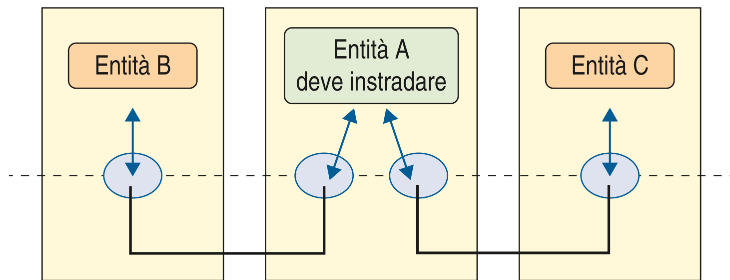


◀ Next hop routing Nelle tabelle di routing per ogni rete di destinazione è indicato solo il prossimo router (next hop) nel percorso verso la destinazione. ▶

Tabella di instradamento o routing

L'instradamento IP si basa su tabelle presenti su host e router che prendono il nome di tabelle di instradamento o routing table.

Le tabelle d'instradamento elencano, per ciascuna sottorete nota, il relativo netID e l'indirizzo IP del router d'inoltro.



Ogni router contiene una tabella di instradamento che contiene informazioni relative alle destinazioni conosciute in modo tale che quando arriva un pacchetto venga instradato verso la destinazione. Una tabella di instradamento è composta da un insieme di righe (entry) ciascuna contenente almeno i seguenti quattro elementi:

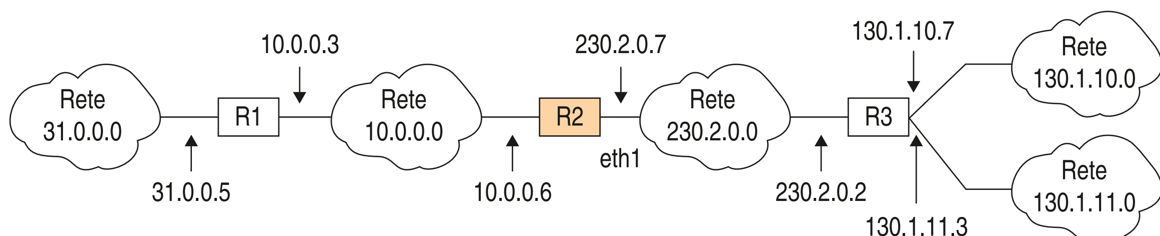
- un indirizzo della rete di destinazione: il router può avere più di un percorso per la stessa destinazione (indicata con *N*);
- la maschera di rete associata all'indirizzo di destinazione (indicata con *M*);
- l'interfaccia su cui inoltrare i pacchetti (indicata con *I*);
- l'indirizzo del prossimo router (next hop router) lungo la strada che porta alla destinazione (indicata con *NH*).

> A questi dati si aggiunge il costo per raggiungere la destinazione sul percorso, che inizia con l'interfaccia indicata nella riga; il costo consentirà all'**IS** di scegliere tra eventuali percorsi alternativi. L'unità di misura di questo costo dipende dal protocollo utilizzato.

Il router non conosce il cammino completo che il datagramma dovrà compiere e la tabella di instradamento specifica quindi solo un passo lungo il cammino verso la destinazione: impropriamente si chiamano genericamente con il termine "route" le informazioni presettate su una riga della tabella di routing che dovrebbero invece chiamarsi *"un hop nella route".*

ESEMPIO 1

Vediamo un esempio con una rete dove sono presenti tre router.



La tabella di instradamento di R2 è la seguente:

Destinazione N	Maschera M	Next hop NH	Interfaccia I
10.0.0.0	255.0.0.0	ind.diretto (10.0.0.6)	ethernet0
230.2.0.0	255.255.0.0	ind.diretto (230.2.0.7)	ethernet1
31.0.0.0	255.0.0.0	10.0.0.3	ethernet0
130.1.10.0	255.255.255.0	230.2.0.2	ethernet1
130.1.11.0	255.255.255.0	230.2.0.2	ethernet1

Nelle prime due righe sono stati inseriti gli indirizzi degli host che sono direttamente raggiungibili dal router in quanto appartengono al segmento di rete al quale appartiene egli stesso: rispettivamente l'host con indirizzo 10.0.0.6 appartiene alla rete 10.0.0.0 e quello con indirizzo 230.2.0.2 alla sottorete 230.2.0.0 e, quindi, per essi avviene un instradamento diretto.
Le altre tre reti vengono raggiunte attraverso i router R1 e R3.

Se una destinazione non è presente nella tabella di routing il pacchetto viene indirizzato al router di default.

Router di default (default gateway)

Il router verso cui è inviato il traffico diretto a una destinazione non presente nella tabella di routing prende il nome di router di default o default gateway: non è obbligatorio indicare questa riga nella tabella ma è caldamente consigliato inserirla in quanto questa alternativa di instradamento viene molto utilizzata sia dagli host che dai router.

L'indirizzo di ◀ **default gateway** ▶ viene inserito come ultima riga della tabella di instradamento ed è codificato mediante tutti zero sia nel campo N che nel campo M.

◀ **Default gateway** The gateway in a network that a computer will use to access another network if a gateway is not specified for use. ▶



Destinazione N	Maschera M	Next hop NH	Interfaccia I
10.0.0.0	255.0.0.0	ind.diretto (10.0.0.6)	ethernet0
230.2.0.0	255.255.0.0	ind.diretto (230.2.0.7)	ethernet1
31.0.0.0	255.0.0.0	10.0.0.3	ethernet0
130.1.10.0	255.255.255.0	230.2.0.2	ethernet1
130.1.11.0	255.255.255.0	230.2.0.2	ethernet1
0.0.0.0	0.0.0.0	10.0.0.3	ethernet0

Il router elabora l'indirizzo di un host per individuare a quale sottorete appartiene come di seguito riportato: indicando con X l'indirizzo dell'host desiderato si cerca quali tra gli indirizzi presenti nella tabella di routing hanno il miglior matching, cioè hanno in comune il maggior numero di bit utilizzando la maschera M della rete di destinazione letta nella tabella.

Vengono effettuate le seguenti operazioni:
- X AND M per estrarre l'indirizzo di sottorete;
- Y AND M anch'essa per estrarre l'indirizzo di rete, e questa operazione viene effettuata per tutti gli indirizzi presenti nella tabella di routing finché si verifica che:

X AND M = Y AND M

Nel caso in cui il matching dà esito positivo per più righe si attua la regola del Longest Prefix Matching, cioè si utilizza la riga che ha il maggior numero di bit in comune con X AND M.

Vediamo un semplice esempio con solo tre righe nella tabella di routing.

ESEMPIO 2

Data la seguente tabella di instradamento:

Destinazione (Y)	M	Porta di uscita
198.12.0.0	255.255.0.0	1
198.12.17.0	255.255.255.0	5
0.0.0.0	0.0.0.0	2

Individuare verso quale porta vengono istradati i seguenti pacchetti:

- **A** indirizzo 198.12.17.3
- **B** indirizzo 198.12.0.3
- **C** indirizzo 20.12.0.3

Soluzione

A 198.12.17.3
- porta 1: matching con prefisso lungo 16;
- porta 5: matching con prefisso lungo 24.

Viene quindi scelto il prefisso più lungo e il pacchetto sarà inoltrato nella porta 5.

B 198.12.0.3
- porta 1: matching prefisso lungo 16;
- porta 5: no matching.

C 20.12.0.3
- porta 1: no matching;
- porta 5: no matching.

Quindi i pacchetti rispettivamente vengono instradati:

- **A** 198.12.17.3 → porta 5 dove abbiamo 24 come lunghezza di matching;
- **B** 198.12.0.3 → porta 1 che in questo caso è l'unica che ha un matching;
- **C** porta 2: in questo caso verrà instradato nella default gateway.

Il router deve sempre compiere queste operazioni ogni qualvolta deve inoltrare un pacchetto per individuare la riga "migliore" che più si avvicina alla sottorete nella quale è presente il destinatario del pacchetto stesso: il tempo di ricerca prende il nome di table lookup e mediamente è pari alla metà del numero di righe. Al crescere della tabella la ricerca della route può introdurre ritardi anche non trascurabili e quindi è importante l'organizzazione della tabella per ridurne la complessità e, dove possibile, il numero di righe, per massimizzare le prestazioni del dispositivo.

ESEMPIO 3

Vediamo un secondo esempio dove è necessario inviare un pacchetto all'indirizzo **131.175.21.77/24** Per prima cosa occorre capire se appartiene alla sottorete di una delle interfacce; per effettuare la verifica si fa un AND bit a bit tra indirizzo dell'interfaccia e *netmask* e tra indirizzo di destinazione e *netmask*, se i due risultati coincidono allora la sottorete è la stessa e si procede all'inoltro diretto.

Se per esempio avessimo un'interfaccia con indirizzo 131.175.21.96/24 si procederebbe nel seguente modo:

destinazione: **(131.175.21.77) AND (255.255.255.0) = 131.175.21.0**
interfaccia x: **(131.175.21.96) AND (255.255.255.0) = 131.175.21.0** → confronto positivo

Se i confronti con le interfacce sono negativi occorre procedere a un inoltro indiretto analizzando riga per riga la tabella di *routing* ed effettuando il confronto allo stesso modo usando la *netmask* relativa a ciascuna riga: se il confronto dà esito positivo per più righe della tabella viene selezionata la tabella con la *netmask* che ha il maggior numero di 1 (si dice comunemente che vale il principio del prefisso più lungo).

Supponiamo di avere tre interfacce eth0, eth1 ed eth2

Interfaccia eth0	
IP address	131.17.123.1
Netmask	255.255.255.0

Interfaccia eth1	
IP address	131.17.78.1
Netmask	255.255.255.0

Interfaccia eth2	
IP address	131.17.15.12
Netmask	255.255.255.0

e la seguente tabella di routing:

Rete	Maschera	First hop
131.175.21.0	255.255.255.0	131.17.123.254
131.175.16.0	255.255.255.0	131.17.78.254
131.56.0.0	255.255.0.0	131.17.15.254
131.155.0.0	255.255.0.0	131.17.15.254
0.0.0.0	0.0.0.0	131.17.123.254

Individuiamo come avviene l'inoltro per pacchetti con indirizzo di destinazione:

1 131.175.21.96

a) analizzando gli indirizzi delle tre interfacce non viene riscontrato un confronto positivo;
b) effettuiamo il confronto con le singole righe della tabella otteniamo:

destinazione: **(131.175.21.77) AND (255.255.255.0) = 131.175.21.0**
riga 1: **(131.175.21.0) AND (255.255.255.0) = 131.175.21.0** ➔ confronto positivo

il pacchetto viene quindi inoltrato verso il router con indirizzo 131.17.123.254;

2 131.17.123.88

a) analizzando gli indirizzi delle tre interfacce partendo da eth0;

destinazione: **(131.17.123.88) AND (255.255.255.0) = 131.17.123.0**
eth0: **(131.17.123.1) AND (255.255.255.0) = 131.17.123.0** ➔ confronto positivo

viene inoltrato sull'interfaccia eth0 mediante il mapping con l'indirizzo MAC;

3 131.56.78.4

a) analizzando gli indirizzi delle tre interfacce non viene riscontrato un confronto positivo;
b) effettuiamo il confronto con le singole righe della tabella e dalla riga otteniamo:

destinazione: **(131.56.78.4) AND (255.255.0.0) = 131.56.0.0**
riga 3: **(131.56.0.0) AND (255.255.0.0) = 131.56.0.0** ➔ confronto positivo

viene inoltrato al gateway 131.17.15.254;

4 190.78.90.2

a) analizzando gli indirizzi delle tre interfacce non viene riscontrato un confronto positivo;
b) effettuiamo il confronto con le singole righe non viene riscontrato un confronto positivo; il pacchetto viene quindi inoltrato verso il **default gateway** con indirizzo 131.17.123.254.

Route a costi diversi

Abbiamo detto che nelle tabelle di routing è presente anche il campo costo la cui unità di misura dipende dal protocollo utilizzato: è possibile che in una tabella siano presenti due righe aventi la medesima destinazione con next hop diverso e costo diverso.

Questa situazione si presenta nel caso di topologie di rete diverse da quelle presentate nei precedenti esempi in quanto è necessaria la presenza di percorsi alternativi verso una certa destinazione e quindi la rete deve avere delle maglie.

Per esempio in una tabella potremmo avere le due righe seguenti:

Destinazione (Y)	M	Next hop	Porta di uscita	Costo
...				
80.12.0.0	255.255.0.0	50.12.0.0	1	3
80.12.0.0	255.255.0.0	70.33.0.0	5	5
.	.		.	

La prima route risulta conveniente rispetto alla seconda e quindi verrà sempre scelta per raggiungere la destinazione 80.12.0.0.
La seconda route rimane però come alternativa nel caso in cui ci fosse un guasto o all'interfaccia o al router del percorso precedente: prende il nome di route di backup.

Viene però raramente utilizzata in quanto non tutti i router si accorgono del malfunzionamento di una interfaccia e, come vedremo, nel caso di routing dinamico il ricalcolo dei percorsi avviene automaticamente e quindi non è necessario avere due alternative per la stessa destinazione.

Vediamo un secondo esempio, dove supponiamo duplicata una destinazione avente interfaccia e lunghezza di maschera diversa:

Destinazione (Y)	M	Next hop	Porta di uscita	Costo
...				
80.12.1.0	255.255.255.0	50.12.0.0	1	5
80.12.1.0	255.255.254.0	70.33.0.0	5	2
.	.		.	

L'algoritmo di instradamento si trova a dover scegliere tra due alternative aventi ciascuna un parametro migliore rispetto all'altra: generalmente la prima scelta avviene sul miglior matching e solo come seconda opzione quella a costo inferiore.

La scelta di una route in base al costo viene effettuata solo come alternativa tra due route aventi lo stesso livello di specifica, cioè la stessa lunghezza di subnet mask.

Aggregazione di indirizzi

Il numero delle righe di una tabella di routing incide sull'efficienza di tutta la rete e, quindi, appena possibile è necessario cercare di renderlo il più piccolo possibile mediante l'aggregazione di route in modo da ridurne il numero.

Per poter aggregare due (o più) route è innanzi tutto necessario che queste condividano lo stesso valore di next hop: inoltre queste route devono essere relative a reti non direttamente connesse dato che le route direttamente connesse non possono essere cancellate dalla routing table.

Vediamo un esempio con la seguente tabella:

Destinazione (Y)	M	Next hop	Costo
80.10.2.0/25	255.255.255.128	80.10.2.1	1
80.10.2.128/30	255.255.255.192	80.10.2.129	1
80.10.2.132/30	255.255.255.192	80.10.2.130	1
80.10.0.0/24	255.255.255.0	80.10.2.130	2
80.10.1.0/24	255.255.255.0	80.10.2.130	2

Abbiamo tre righe (le ultime tre) che condividono lo stesso next hop e sono quindi le sole candidate alla aggregazione, ma l'aggregazione della terza riga con le altre implicherebbe un notevole peggioramento del matching mentre le ultime due righe, avendo la stessa subnet mask, introducono solamente l'allungamento unitario: vengono raggruppate in

80.10.0.0/23	255.254.0.0	80.10.2.130	2

e la tabella si riduce quindi alla seguente:

Destinazione (Y)	M	Next hop	Costo
80.10.2.0/25	255.255.255.128	80.10.2.1	1
80.10.2.128/30	255.255.255.192	80.10.2.129	1
80.10.2.132/30	255.255.255.192	80.10.2.130	1
80.10.0.0/23	255.255.254.0	80.10.2.130	2

Se avessimo raggruppato le ultime tre righe avremmo ottenuto questa situazione:

Destinazione (Y)	M	Next hop	Costo
80.10.2.0/25	255.255.255.128	80.10.2.1	1
80.10.2.128/30	255.255.255.192	80.10.2.129	1
80.10.0.0/22	255.255.252.0	80.10.2.130	2

La scelta di raggruppare più o meno righe creando di fatto supernet molto ampie come nel secondo esempio oppure di preferire l'unificazione delle route simili peggiorando il meno possibile il matching (come vedremo nelle routing statico), è lasciato all'operatore.

Il caso estremo è quello riportato nell'esempio seguente:

Destinazione (Y)	M	Next hop	Costo
80.10.2.0/25	255.255.255.128	80.10.2.1	1
80.10.2.128/30	255.255.255.192	80.10.2.129	1
80.10.2.132/30	255.255.255.192	80.10.2.130	1
80.10.1.0/24	255.255.255.0	80.10.2.133	2
80.10.1.128/30	255.255.255.252	80.10.2.133	1

L'aggregazione delle due ultime viene fatta sostituendole con la route di default:

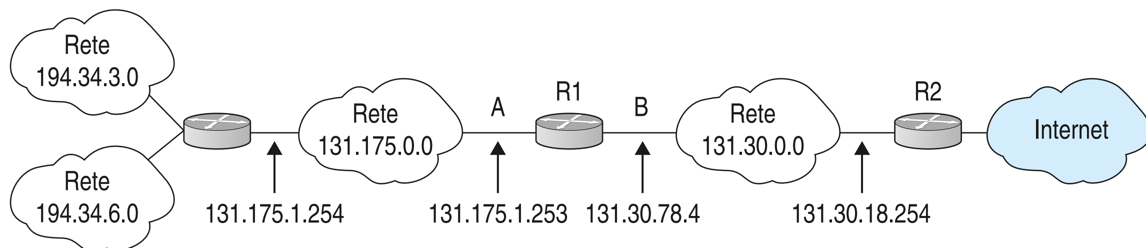
0.0.0.0	0.0.0.0	80.10.2.133	2

Verifichiamo le conoscenze

Esempio svolto

Vediamo un semplice esempio.
Data la rete di figura, scrivi la tabella di instradamento del router R1 per indirizzare anche le seguenti reti:

194.34.3.0/24
194.34.6.0/24
140.56.0.0/16
141.56.0.0/16



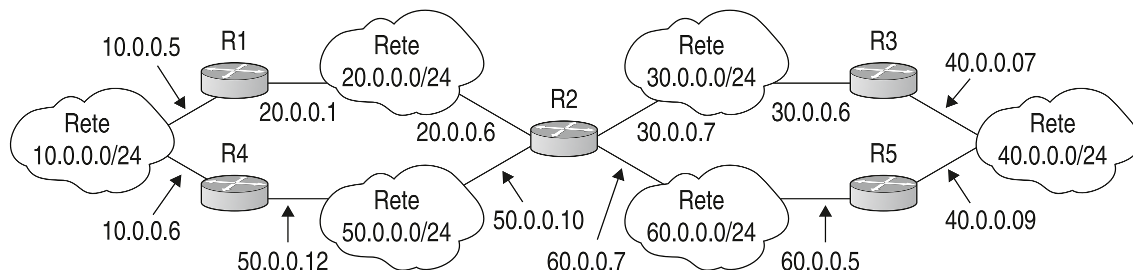
Soluzione

Le reti 140.56.0.0 141.56.0.0 non sono presenti nella topografia raffigurata e quindi possiamo pensare che siano reti presenti in internet: la tabella di instradamento del router R1 è quindi la seguente:

Destinazione	Next hop
194.34.23.0	131.175.1.254
194.34.34.0	131.175.1.254
140.56.0.0	131.30.18.254
141.56.0.0	131.30.18.254
131.175.0.0	interface A
131.30.0.0	interface B
...	...
default	131.30.18.254

Esempio svolto parzialmente

Dato il seguente sistema scrivi le routing table per i router R2 e R3.



Soluzione

Routing Table di R2	
Net_Id	**Router_Id**
10.0.0.0./24	20.0.0.1
20.0.0.0./24	Instradamento diretto
30.0.0.0./24	Instradamento diretto
40.0.0.0./24	30.0.0.6
50.0.0.0./24	Instradamento diretto
60.0.0.0./24	Instradamento diretto

Routing Table di R3	
Net_Id	**Router_Id**
10.0.0.0./24	
20.0.0.0./24	
30.0.0.0./24	
40.0.0.0./24	
50.0.0.0./24	
60.0.0.0./24	

Esercizi

1 Si consideri la rete di figura in cui siano NA=118, NB=28, NC=58, ND=5 rispettivamente il numero di host delle sottoreti A, B, C, D.

Nell'ipotesi che all'amministratore della rete sia assegnato il blocco di indirizzi di classe C 193.101.66.0/24 completa la tabella di instradamento del router R riportando l'indirizzo e la maschera di ciascuna rete applicando la tecnica del subnetting e minimizzando il numero di indirizzi IP utilizzati.

Completa la tabella indicando l'identificativo di interfaccia del router verso cui vanno instradati i pacchetti diretti verso le sottoreti.

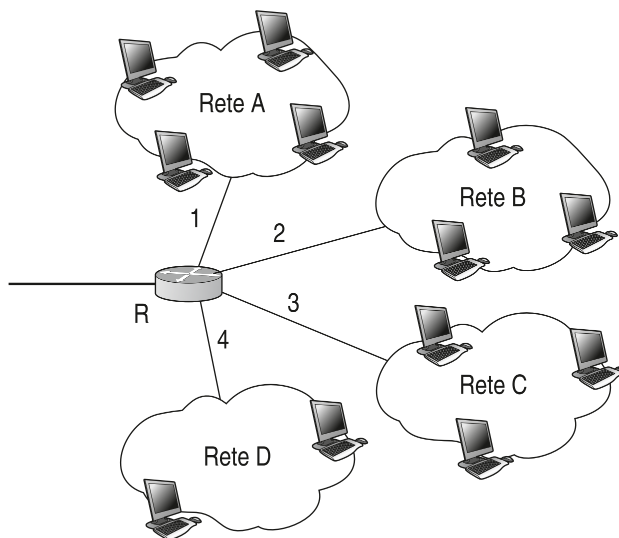
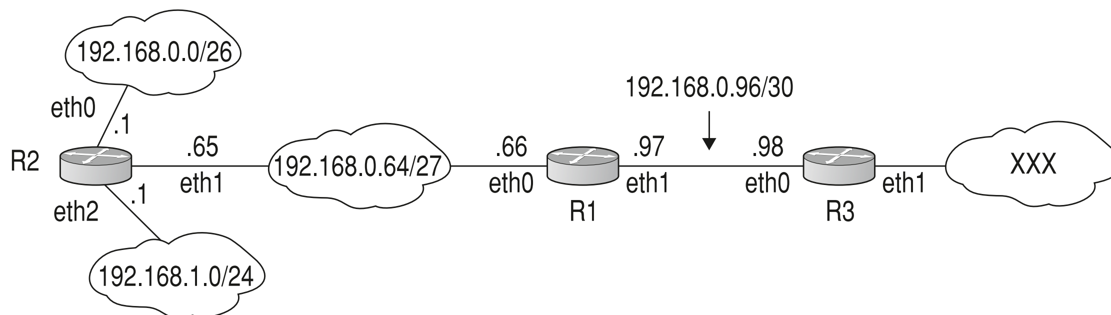


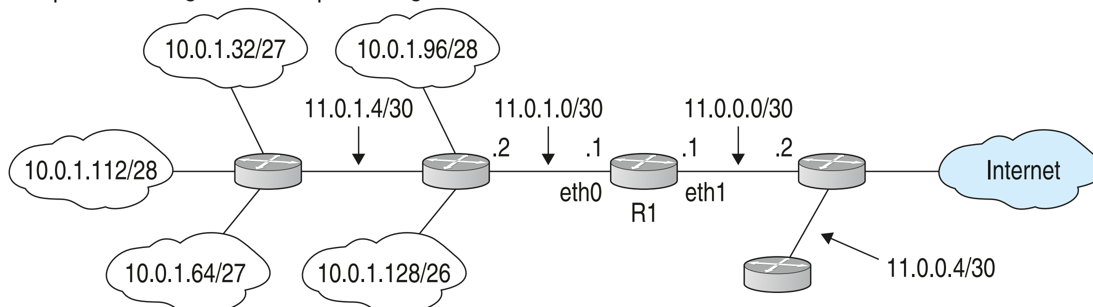
Tabella di instradamento di R		
Indirizzo di sottorete	**Maschera di sottorete**	**Identificativo interfaccia**

2 Data la rete in figura con l'indirizzamento indicato scrivi la tabella di routing del router R1.



Destinazione	Maschera	Next hop	Interfaccia
192.168.0.64			
192.168.0.96			
192.168.0.0			
192.168.1.0			
0.0.0.0	0.0.0.0		

3 Data la rete in figura con l'indirizzamento indicato scrivi la tabella di routing del router R1 quindi raggruppa ove possibile le righe in una supernetting in modo da ridurre al massimo la dimensione della tabella.

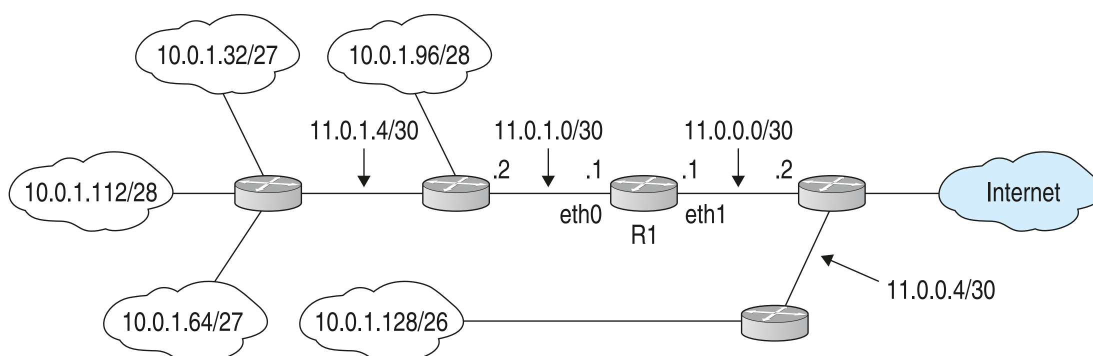


Destinazione	Maschera	Next hop	Interfaccia

Dopo aver raggruppato le .... righe con .... la tabella diviene:

Destinazione	Maschera	Next hop	Interfaccia

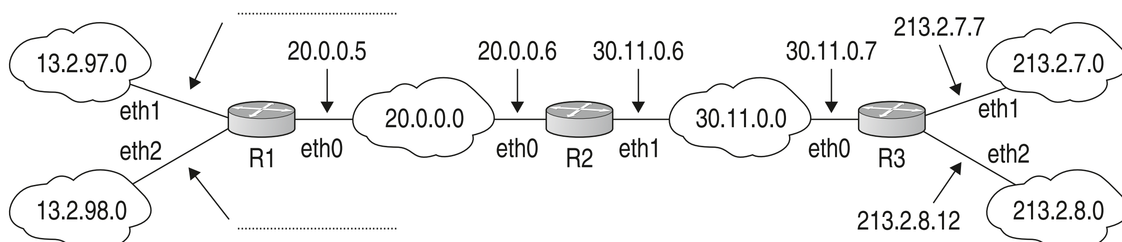
4 Modifica la routing table dell'esercizio precedente nel caso in cui la rete 10.0.1.128 diventa accessibile tramite un diverso percorso di rete, come indicato nella seguente figura.



Destinazione	Maschera	Next hop	Interfaccia

5 Data la rete di figura

a) assegna i due indirizzi mancanti alle interfacce del router R1;
b) completa le tabelle di routing per i tre router.



Router R1

Destinazione N	Maschera M	Next hop NH	Interfaccia I
13.2.97.0			
13.2.98.0			
20.0.0.0			
30.11.0.0			
210.2.7.0			
210.2.8.0			

Router R2

Destinazione N	Maschera M	Next hop NH	Interfaccia I
13.2.97.0			
13.2.98.0			
20.0.0.0			
30.11.0.0			
210.2.7.0			
210.2.8.0			

Router R3

Destinazione N	Maschera M	Next hop NH	Interfaccia I
13.2.97.0			
13.2.98.0			
20.0.0.0			
30.11.0.0			
210.2.7.0			
210.2.8.0			

6 Un router ha la seguente tabella di routing e la seguente configurazione delle interfacce.

Rete	Maschera	First hop
131.175.15.0	255.255.255.0	131.175.21.1
131.175.16.0	255.255.255.0	131.175.21.2
131.175.17.0	255.255.255.0	131.175.21.3
131.180.23.0	255.255.255.0	131.175.21.4
131.180.18.0	255.255.255.0	131.175.21.4
131.180.21.0	255.255.255.0	131.175.21.4
131.180.0.0	255.255.255.0	131.175.21.5
0.0.0.0	0.0.0.0	0.0.0.0

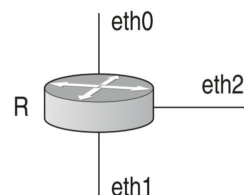
interfaccia 1: 131.175.21.254, 255.255.255.0
interfaccia 2: 131.175.12.254, 255.255.255.0

Descrivi come avviene l'inoltro per pacchetti con indirizzo di destinazione:
a) 131.175.21.86
b) 131.175.16.65
c) 131.180.21.78
d) 200.45.21.84

a) 131.175.21.86
viene inoltrato .... dato che ....
b) 131.175.16.65
viene inoltrato .... dato che ....
c) 131.180.21.78
viene inoltrato .... dato che ....
d) 200.45.21.84
viene inoltrato .... dato che ....

7 Un router ha la seguente configurazione delle interfacce

Interfaccia	Indirizzo IP	Maschera
Eth0	131.175.123.235	255.255.255.0
Eth1	131.175.123.129	255.255.255.128
Eth2	131.175.122.0	255.255.255.0



e la seguente tabella di routing

Rete	Maschera	Next hop
130.170.0.0	255.255.0.0	131.175.123.1
130.171.0.0	255.255.0.0	131.175.123.132
130.171.4.0	255.255.252.0	131.175.122.2
130.170.10.0	255.255.254.0	131.175.122.3
0.0.0.0	0.0.0.0	131.175.123.3

Il router riceve gli 8 pacchetti riportati di seguito, per ciascuno dei quali vengono riportati l'indirizzo IP di destinazione e l'interfaccia attraverso cui il router riceve il pacchetto.

Pacchetti ricevuti

A. 131.175.123.64 da eth2
B. 131.175.123.255 da eth0
C. 131.175.123.132 da eth2
D. 130.170.132.240 da eth1
E. 130.170.11.64 da eth1
F. 130.171.5.125 da eth1
G. 156.198.34.14 da eth0
H. 0.0.0.132 da eth1

Si chiede di indicare il comportamento del router per ciascuno dei pacchetti specificando se il router scarta o inoltra il pacchetto. Nel caso in cui il router decida di inoltrare il pacchetto, specifica l'indirizzo IP del next hop e se l'inoltro è di tipo diretto o indiretto.

A. 131.175.123.64 ....

B. 131.175.123.255 ....

C. 131.175.123.132 ....

D. 130.170.132.240 ....

E. 130.170.11.64 ....

F. 130.171.5.125 ....

G. 156.198.34.14 ....

H. 0.0.0.132 ....

Lezione 2

ROUTING STATICO E DINAMICO

IN QUESTA LEZIONE IMPAREREMO...

- la differenza tra routing statico e dinamico
- le politiche di instradamento
- i protocolli per il routing distribuito

Routing statico e routing dinamico

Si è detto che il routing è il processo usato dal router per individuare l'interfaccia di uscita su cui rilanciare i pacchetti verso la destinazione che viene individuata in base al valore dell'IP address confrontandolo con i dati presenti nella tabella di routing.

Lo scopo ultimo di un protocollo di routing consiste nel creare una tabella di instradamento in ciascun nodo della rete in modo che esso possa prendere la decisione locale sulla base della conoscenza dello stato dell'intera rete: questa è la *"difficoltà principale"* del routing.

> Nella tabella sono presenti i *possibili percorsi* verso le reti remote: questi dati devono essere inseriti e memorizzati nel dispositivo e mantenuti costantemente aggiornati.

Esistono due modalità per creare e gestire le tabelle di routing:
- ◀ **routing statico (static routing)** ▶: la configurazione viene effettuata dall'amministratore della rete;
- ◀ **routing dinamico (dynamic routing)** ▶: le informazioni vengono ricevute dagli altri router.

Nel routing statico le operazioni possono essere divise in 3 parti:
1. l'amministratore di rete individua la route manualmente;
2. "installa" questa route nelle routing table;
3. i pacchetti sono inoltrati usando la route statica.



◀ **Static routing** Use a programmed route that a network administrator enters into the router. **Dynamic routing** Use a route that a routing protocol adjusts automatically for topology or traffic changer. ▶

Questa non è una ◀ tecnica scalabile ▶ e in caso di modifiche nella topologia di rete è necessario che l'amministratore apporti sempre manualmente le opportune variazioni alla configurazione delle route statiche: di fatto questo è un problema nella gestione di reti di grandi dimensioni.
Gli algoritmi che implementano il routing statico sono di tipo non adattivo in quanto le decisioni di routing sono prese in anticipo, all'avvio della rete, e comunicate ai router che poi vi si attengono sempre.



◀ Sistema scalabile Un sistema è scalabile se può essere adattato a diversi contesti con forti differenze di complessità senza che questo richieda la riprogettazione dello stesso sistema. ▶

Nel routing dinamico le decisioni di routing sono riformulate molto spesso in quanto le tabelle sono in continuo aggiornamento in tempo reale in base al traffico, alla topologia della rete ecc.

Politiche di instradamento (o algoritmi di instradamento)



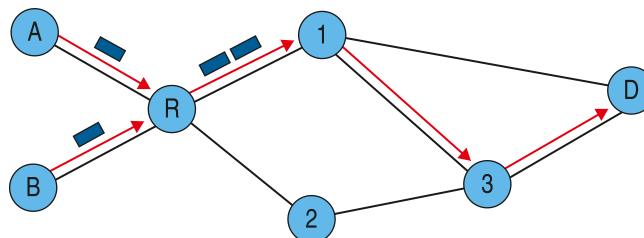
POLITICA DI ROUTING

La politica di *routing* (o *algoritmo di routing*) è quella che definisce i criteri di scelta del cammino nella rete per i pacchetti che viaggiano tra un nodo di ingresso e uno di uscita: è dunque quella che costruisce le tabelle di *routing* che viene usata dai nodi per effettuare il *forwarding.*

Il tipo di inoltro (forwarding) utilizzato dalle reti IP condiziona la scelta delle politiche di routing: ricordiamo che il forwarding IP è basato sull'indirizzo di destinazione (destination-based) con inoltro al nodo successivo (next hop routing) e quindi, come conseguenza, i pacchetti diretti a una stessa destinazione **D** che giungono in un router **R** seguono lo stesso percorso da **R** verso **D** indipendentemente dal link di ingresso in **R.**

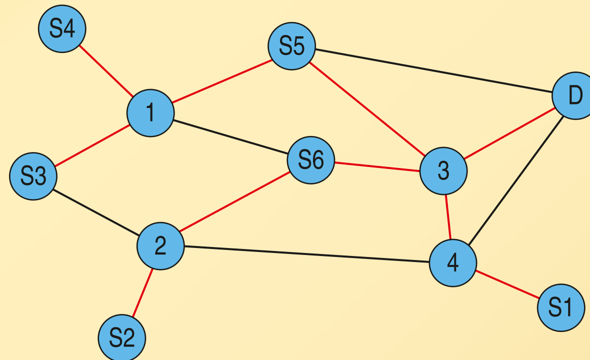
ESEMPIO 4

Supponiamo di avere la rete di figura: al router R arrivano pacchetti sia dal nodo A che dal nodo B, con destinazione D.
Il nodo R inoltra tutti i pacchetti che hanno destinazione il nodo D verso il nodo 1 che successivamente provvederà a trasmetterli al nodo 3 e infine giungeranno al nodo D.



La scelta di inoltro verso il nodo 1 piuttosto che verso il nodo 2 viene fatta in base alle politiche di instradamento utilizzando gli algoritmi di routing.

Il vincolo che ogni politica di routing deve soddisfare è che:
l'insieme dei cammini da ogni sorgente verso una destinazione D deve essere un albero (quindi senza percorsi ciclici) per ogni possibile destinazione D, come nella seguente figura:



In questo caso da ogni nodo sorgente S2, S3, S4, S5, S6 deve essere presente un cammino che raggiunge il nodo D e ogni pacchetto segue sempre lo stesso percorso fino a che non vengono cambiate le tabelle di instradamento presenti nei router intermedi.

Il percorso sorgente-destinazione deve essere determinato considerando quindi l'intero cammino da percorrere e la politica di routing utilizzata sin dall'introduzione delle reti TCP/IP è basata sul calcolo dei cammini minimi.

Come vedremo nelle prossime lezioni tale calcolo è effettuato sul grafo che rappresenta la rete nel quale a ogni arco è associato un peso opportunamente scelto (metrica) e su di esso vengono applicati algoritmi classici della teoria dei grafi che permettono di individuare i cammini minimi generando l'albero dei cammini minimi.

Le politiche che permetto di realizzare l'instradamento all'interno delle reti sono tra loro differenti in base a quando ricevono le informazioni, come ricevono le informazioni, a quanto spesso rivedono le loro decisioni e alla metrica di valutazione che adottano: in base a queste caratteristiche vengono classificati in politiche:

- isolate: calcolano il routing con sole informazioni locali, indipendentemente dallo stato degli altri nodi e dallo stato della intera rete;
- centralizzate: è quello più semplice, ma non scalabile in quanto un centro di controllo conosce lo stato globale e calcola il cammino ottimo per ogni coppia (mittente, destinatario) e dirama le tabelle (per esempio TYMNET);
- miste: si uniscono i vantaggi delle precedenti, combinando politiche isolate e centralizzate (per esempio TRANSPAC);
- distribuite: è quello più complesso, ma è scalabile e robusto; in esso i nodi cooperano e comunicano frequentemente il proprio stato e quello della rete (come per esempio in Internet).

Routing distribuito

Definzioni

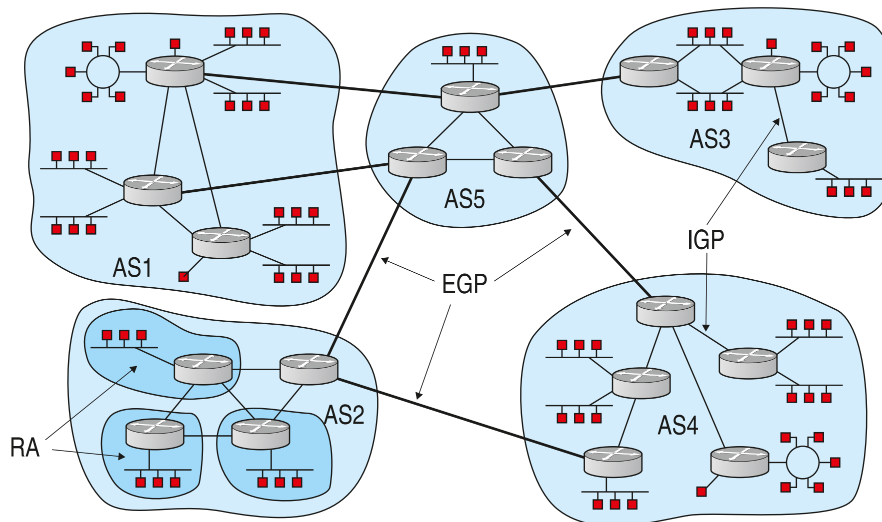
Le reti WAN (o GAN come Internet) sono una aggregazione di LAN (chiamati anche Sistemi Autonomi, Autonomous System AS) aventi ciascuna una propria politica di gestione e di amministrazione, che però devono interagire tra loro.
Per ogni AS possiamo definire un sistema interno con un ben determinato confine e un interfacciamento verso il sistema esterno.

Un ◀ **Autonomous System** ▶ può essere anch'esso suddiviso in un insieme di Routing Area (RA) interconnesse da un backbone (dorsale) e in questo caso ogni network IP è tutta contenuta in una RA.



◀ **Autonomous System (AS):** is a collection of connected Internet Portocol (IP) routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet. ▶

I protocolli utilizzati dai router presenti nel sistema interno sono chiamati Interior Gateway Protocol (IGP) mentre i protocolli utilizzati per l'interconnessione sono gli Exterior Gateway Protocol (EGP).



Internet non è altro che un insieme di AS interconnessi.

Protocolli per il routing distribuito

Nel routing distribuito ogni router calcola le sue tabelle dialogando con gli altri router: ogni router informa gli altri riguardo le "rotte" che conosce tramite dei protocolli ausiliari di livello rete che vedremo in seguito.



◀ **Link state** A link-state routing protocol is one of the two main classes of routing protocols used in packet switching networks: the basic concept of link-state routing is that every node constructs a *map* of the connectivity to the network showing which nodes are connected to which other nodes. ▶

Negli algoritmi che implementano queste politiche possiamo individuare due approcci principali:

A algoritmi statici o con conoscenza completa (di ◀ **link state** ▶): ogni router contiene la topologia della rete e le informazioni sui "suoi vicini" vengono mandate a tutti gli altri router con un messaggio di broadcast (*Link State Packet*) solo quando ci sono stati dei cambiamenti della topologia; dato che ciascun router possiede informazioni sulla topologia di rete completa è in grado di calcolare in modo indipendente il miglior hop successivo per ogni possibile destinazione della rete locale.

Gli algoritmi sono di tre tipi:
- shortest path routing;
- flooding;
- flow-based routing.

Tra questi analizzeremo in dettaglio, nella lezione 3, l'algoritmo Shortest Path First (SPF) implementato con l'algoritmo di Dijkstra.

Ⓑ algoritmi dinamici o con conoscenza parziale (di ◀ **distance vector** ▶): in questi protocolli ciascun router non dispone di informazioni sulla topologia di rete completa ma dialoga con gli altri router e riceve informazioni che gli permettono di popolare la tabella raccogliendo i dati sugli aggiornamenti delle configurazioni delle diverse sottoreti: il processo di aggiornamento continua finché non si giunge ad avere delle informazioni stabili in ciascun router.

Tra questi noi analizzeremo in dettaglio l'algoritmo di Bellman-Ford nella prossima lezione.



◀ **Distance vector** A distance vector routing protocol is one of the two main classes of routing protocols used in packet switching networks: requires that a router informs its neighbors of topology changes periodically. ▶



Zoom su...

PROTOCOLLI IGP E EGP

I protocolli utilizzati per sistemi interni, cioè gli IGP, sono:
- Distance-vector routing protocol
  - Routing Information Protocol (RIP)
  - Interior Gateway Routing Protocol (IGRP)
  - Enhanced Interior Gateway Routing Protocol (EIGRP)
- Link-state routing protocol
  - Open Shortest Path First (OSPF)
  - Intermediate System To Intermediate System (IS-IS)

Per l'interconnessione tra sistemi in Internet si utilizza un unico protocollo, il Border Gateway Protocol Version 4 (BGP-4) definito in RCF 1771 che è quindi l'unico protocollo EGP.

Scelta dell'algoritmo di routing

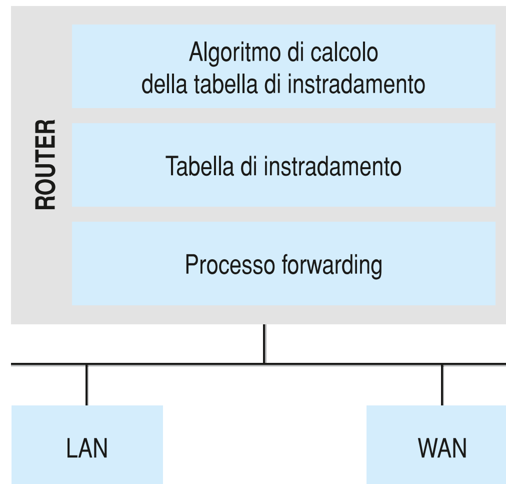
Nella scelta dell'algoritmo di routing possono esistere più criteri di ottimalità contrastanti: se da una parte una esigenza è quella di "minimizzare il ritardo medio di ogni pacchetto" dall'altra si cerca di "massimizzare l'utilizzo dei link della rete"; quindi un algoritmo che opera su un grande numero di nodi potrebbe richiedere tempi di calcolo inaccettabili.

I parametri sui quali avviene la scelta dell'algoritmo di routing sono i seguenti:
- semplicità: i router hanno CPU e memoria finite;
- robustezza: devono essere in grado di adattarsi alle variazioni di topologia e di carico sulla rete;
- stabilità: l'algoritmo deve convergere in tempo accettabile e produrre un risultato utilizzabile;

- equità: deve garantire lo stesso trattamento a tutti i nodi della rete;
- metrica da adottare: per esempio il numero di salti effettuati, somma dei costi di tutte le linee attraversate, ecc.

Nella maggior parte delle reti i due tipi di routing, statico e dinamico, operano congiuntamente.

Possiamo schematicamente rappresentare le attività di un router come nella seguente figura:



Il compito da svolgere, cioè il processo principale del router, è il ◀ forwarding ▶ per i pacchetti che viaggiano tra un nodo di ingresso e uno di uscita: questo viene realizzato mediante politiche di routing che definiscono i criteri di scelta del cammino costruendo le tabelle di routing mediante gli algoritmi di instradamento.



◀ Forwarding (inoltro) Trasferisce i pacchetti in ingresso a un router verso l'uscita appropriata del router stesso. ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 **Lo scopo ultimo di un protocollo di routing consiste:**

a) nell'individuare gli host adiacenti
b) nel creare una tabella di instradamento
c) nell'inviare i datagrammi nel nodo più vicino
d) nell'inviare i pacchetti nel nodo più vicino

2 **La politica di *routing* (o *algoritmo di routing*) è quella che:**

a) definisce i criteri di scelta del cammino nella rete
b) definisce le modalità di connessione dei router nella rete
c) definisce come connettere i nodi per ottenere il miglior cammino nella rete
d) nessuna delle precedenti

3 **L'albero dei cammini minimi:**

a) si ottiene dalla somma dei singoli percorsi minimi negli AS
b) si ottiene dal calcolo dei cammini minimi su un grafo pesato per ogni AS
c) si ottiene dal calcolo dei cammini minimi su un grafo pesato di tutta la rete
d) nessuna delle precedenti

4 **Le politiche che permettono di realizzare l'instradamento all'interno delle reti si differenziano (indicare quella errata):**

a) in base a quando ricevono le informazioni
b) a come ricevono le informazioni
c) a dove vengono inviate le informazioni
d) a quanto spesso rivedono le loro decisioni
e) alla metrica di valutazione che adottano

5 **Le politiche di routing sono classificate in (indicare quelle errate):**

a) isolate
b) centralizzate
c) autonome
d) miste
e) interne
f) distribuite

6 **Per ogni Autonomous System AS possiamo definire (3 risposte):**

a) un sistema interno
b) un router autonomo
c) un confine
d) un interfacciamento verso il sistema esterno

7 **Negli algoritmi che implementano queste politiche possiamo individuare due approcci principali:**

a) algoritmi statici o con conoscenza completa
b) algoritmi statici o con conoscenza parziale
c) algoritmi dinamici o con conoscenza completa
d) algoritmi dinamici o con conoscenza parziale

8 **I parametri sui quali avviene la scelta dell'algoritmo di routing sono i seguenti (indica quello errato):**

a) semplicità
b) velocità
c) stabilità
d) equità
e) metrica da adottare

Esercizi di completamento

1 Esistono due modalità per creare e gestire le tabelle di routing:
routing statico: la configurazione ....;
routing dinamico: le informazioni ....

2 Un sistema è scalabile se può essere adattato a diversi contesti con forti differenze di complessità senza che questo richieda la ....

3 Il tipo di inoltro utilizzato dalle reti IP condiziona la scelta delle politiche di routing: il forwarding IP è basato su .... con inoltro ....

4 Le politiche che permetto di realizzare l'instradamento all'interno delle reti sono tra loro differenti in base a .... ricevono le informazioni, ... ricevono le informazioni, a .... rivedono le loro decisioni e alla .... di valutazione che adottano.

5 Nel routing .... ogni router calcola le sue tabelle dialogando con gli altri router: ogni router informa gli altri riguardo le "rotte" che conosce tramite dei .... di livello ....

6 Nella scelta dell'algoritmo di routing possono esistere più criteri di ottimalità contrastanti: se da una parte una esigenza è quella di "minimizzare il ...." dall'altra si cerca di "massimizzare ....".

7 Completa la seguente tabella indicando con una X la classificazione dei protocolli:

	Distance-vector	Link-state	IGP	EGP
Open Shortest Path First (OSPF)				
Intermediate System To Intermediate System (IS-IS)				
Routing Information Protocol (RIP)				
Border Gateway Protocol Version 4 (BGP-4)				
Enhanced Interior Gateway Routing Protocol (EIGRP)				
Interior Gateway Routing Protocol (IGRP)				

Test vero/falso

1 Nel routing statico la configurazione viene effettuata dall'amministratore della rete. V F
2 Nel routing dinamico le informazioni vengono ricevute dagli altri router. V F
3 Nel routing statico le tabelle non subiscono mai modifiche. V F
4 Il routing statico è una tecnica scalabile. V F
5 Nel routing statico gli algoritmi sono di tipo adattativo. V F
6 Il forwarding IP è di tipo destination-based con next hop routing. V F
7 L'insieme dei cammini da ogni sorgente verso una destinazione deve essere un albero binario. V F
8 Una politica mista è quella più complessa, ma è scalabile e robusta (per esempio in internet). V F
9 Internet non è altro che un insieme di AS interconnessi. V F
10 Gli algoritmi statici sono anche detti algoritmi di link state. V F
11 Gli algoritmi dinamici sono anche detti algoritmi di distance state. V F
12 L'algoritmo di Bellman-Ford viene utilizzato come algoritmio di distance vector. V F

Lezione 3

RETI, GRAFI E ALBERI

IN QUESTA LEZIONE IMPAREREMO...

- l'analogia tra reti e grafi
- la ricerca del cammino minimo (shortest path)
- la relazione tra grafi, alberi e spanning tree ottimo

Introduzione

In una rete un host è connesso direttamente al cosiddetto router predefinito (default router) presente nel suo segmento di rete: questo router è anche chiamato router di primo hop e a esso trasferisce tutti i pacchetti che deve spedire.
Noi chiameremo router origine (o di primo hop) il router di default dell'host mittente e router destinazione il router predefinito a cui è connesso l'host di destinazione.
Il problema da risolvere, che è quello di instradare un pacchetto tra host di origine e host di destinazione, si riconduce chiaramente al problema d'instradare il pacchetto tra questi due router.

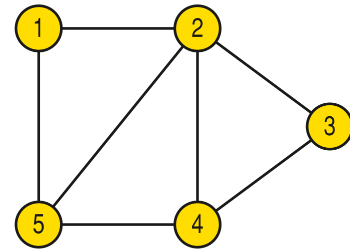
> Effettuare l'istradamento di un pacchetto in una rete equivale a individuare un "percorso" tra sorgente e destinazione: inoltre il cammino ricercato deve essere il più corto possibile, cioè siamo alla ricerca di un cammino minimo.

Concettualmente possiamo indentificare una rete di calcolatori con una struttura dinamica informatica (o matematica) particolare, il grafo e, quindi, effettuare la ricerca del cammino minimo tra due router equivale a effettuare la ricerca di un cammino minimo in un grafo:
- ricerca del cammino a minima distanza, cioè minimo numeri di archi (hop), se il grafo è non pesato;
- ricerca del cammino a minima lunghezza, nel caso di grafo pesato.

In questa lezione analizzaremo le analogie tra le reti e la teoria matematica dei grafi, riprendendo le definizioni e i concetti generali per poi applicarli ai protocolli di routing.

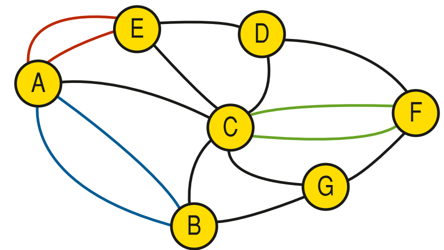
> Come vedremo in seguito, il peso di un grafo rappresenta "la distanza" presente tra due nodi e la lunghezza di un percorso si ottiene dalla somma di tali valori: se non viene esplicitamente indicato, si assume che ogni distanza tra due router abbia valore unitario e quindi la distanza tra origine e destinatario è semplicemente il numero di archi intermedi.

Richiami di matematica discreta: i grafi



Un grafo viene rappresentato genericamente con la schematizzazione illustrata nella figura riportata a lato, costituita da un insieme di vertici (o nodi), connessi tra loro mediante archi (chiamati anche lati o spigoli).

In un grafo non è esclusa la presenza tra due vertici di più di un lato, come nell'esempio seguente, dove le coppie di vertici A-B, A-E e C-F sono connesse con due archi.



Se ora "guardiamo" il grafo come fosse una rete possiamo dire che:
- i vertici sono i router oppure le reti stesse (vedremo in seguito gli AS);
- gli archi:
  – tra due router: sono la connessione punto-punto tra i due router;
  – tra router e rete: rappresentano l'interfaccia che il router ha sulla rete.

La definizione matematica "rigorosa" di grafo è la seguente:



GRAFO

Un grafo G è una coppia (V, E) dove:
- V è l'insieme dei vertici (o nodi);
- E è un insieme di coppie di vertici, detti archi (o spigoli), in cui ogni arco connette due vertici.

Vediamo alcuni esempi di come possono essere costituiti l'insieme V e l'insieme E:
V = {persone che vivono in Italia}, E = {coppie (x, y) tali che *x* ha inviato una mail a *y*};
V = {persone che vivono in Italia}, E = {coppie di persone che si sono strette la mano};
V = {città d'Italia}, E = {coppie di città connesse attraverso una rete ferroviaria};
V = {città d'Italia}, E = {coppie di città connesse attraverso un'autostrada}.

Osservando questi esempi è facile notare le differenze tra il primo esempio e i successivi: nel primo caso, la relazione tra le coppie di elementi non è simmetrica, a differenza delle altre tre situazioni.

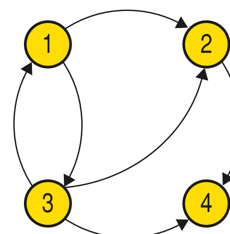


GRAFO ORIENTATO

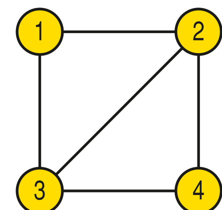
Un grafo orientato (o diretto, chiamato anche con la contrazione digrafo) G è una coppia (V, E) in cui V è un insieme finito, detto insieme dei vertici ed E è una relazione binaria su V che definisce l'insieme degli archi, che sono coppie ordinate di vertici.
Un grafo non orientato è un grafo in cui gli archi sono coppie non ordinate di vertici, cioè un arco tra i due generici vertici *u*, *v* è un insieme di due elementi {*u*, *v*} anziché una coppia (*u*, *v*).

Dal punto di vista grafico, gli archi di un grafo orientato vengono contrassegnati con una freccia che riflette l'ordinamento della coppia di vertici, cioè che va dal nodo di partenza al nodo di arrivo della coppia di nodi.

Quindi in un grafo orientato a ciascun lato è associata una direzione ed è possibile percorrere ciascun



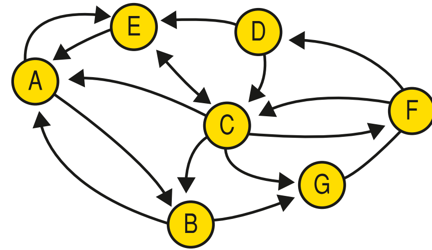
Grafo orientato

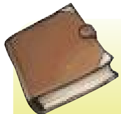


Grafo non orientato

lato da un vertice a un altro in una sola direzione: si può anche ricondurre un grafo non orientato a un grafo orientato sostituendo ogni lato con due lati orientati in senso inverso.

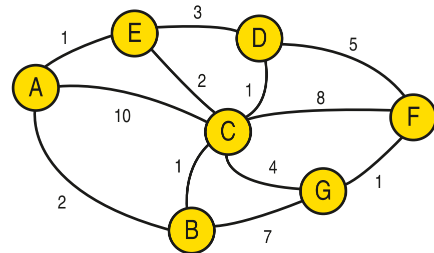
In generale, se non viene esplicitamente indicato, un grafo si intende sempre non orientato.





GRAFO PESATO

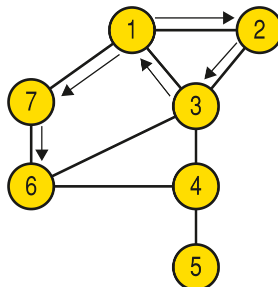
Un grafo pesato è un grafo nel quale viene associata per ogni arco una funzione, che ne calcola il peso; per esempio, in una carta stradale il peso può essere rappresentato dai chilometri di distanza tra due città corrispondenti a due vertici del grafo.

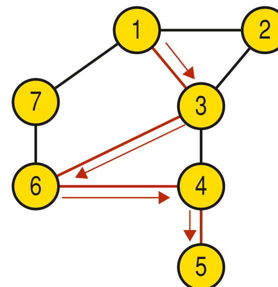




CAMMINO

Con cammino (passeggiata) si intende una sequenza di vertici adiacenti (sia i lati che i vertici).
Un cammino $p = (v_0,v_1,v_2,\ldots,v_k)$ è semplice se contiene solo nodi distinti, cioè $v_i \neq v_j$ per ogni $i \neq j$.

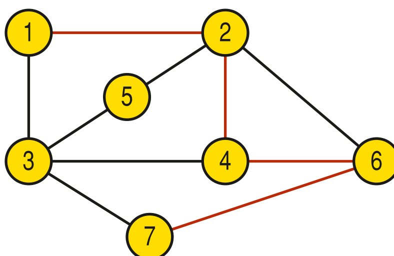


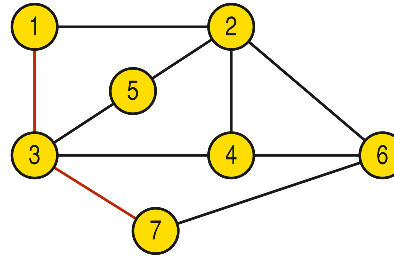


p_1= (1,2,3,1,7,6) è un cammino di lunghezza 5 (il nodo 1 è presente due volte)

p_2= (1,3,6,4,5) è un cammino semplice di lunghezza 5 (tutti i nodi sono presenti una sola volta)

In un grafo, due vertici possono essere uniti da diversi cammini dove con lunghezza di un cammino si intende il numero degli archi che lo compongono: per esempio, nelle due figure qui sotto i cammini <1-2-4-6-7> e <1-3-7> uniscono entrambi i vertici 1 e 7, ma sono di lunghezza diversa.







DISTANZA

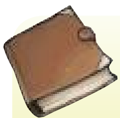
Si definisce distanza la lunghezza del cammino più breve tra due vertici.

Se il grafo è orientato, si deve tener conto della direzione dei lati (non è possibile percorrere un lato dal nodo H al nodo K se l'unico lato che li collega è orientato da K a H).



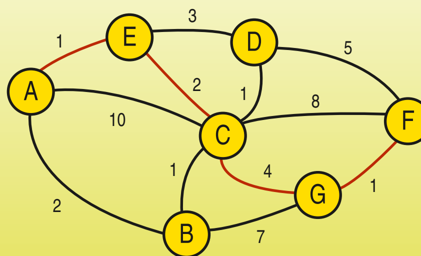
RAGGIUNGIBILITÀ

Dato un grafo G che contiene i vertici A e Z, Z è raggiungibile da A se e solo se esiste nel grafo G un cammino da A a Z.



PESO DI UN CAMMINO

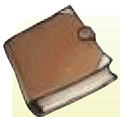
Il peso di un cammino è la somma dei pesi associati a tutti i lati che compongono il cammino.



p_1(A…F) = (A,E,C,G,F)
Peso di p_1 = 1+2+4+1 = 8

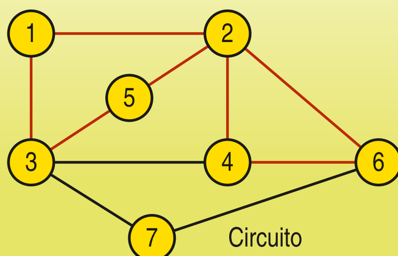
Indichiamo con $w(p)$ la somma dei pesi dei diversi archi del cammino, cioè

$$w(p) = \sum_{(i,j)\in p} w(i,j)$$

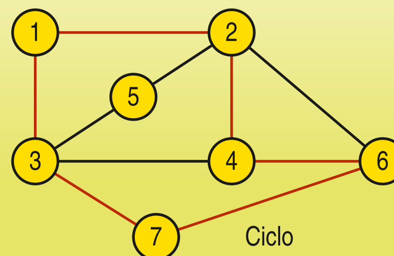


CIRCUITI E CICLI

Un circuito è un particolare cammino che partendo da un nodo v_1 ritorna in v_1 senza archi ripetuti (può anche toccare lo stesso nodo più di una volta).
Un ciclo è un particolare cammino che partendo da un nodo v_1 ritorna in v_1 senza archi né nodi ripetuti: è cioè un cammino $<v_0,v_1,...,v_k>$; è un ciclo se $v_0=v_k$.



Circuito



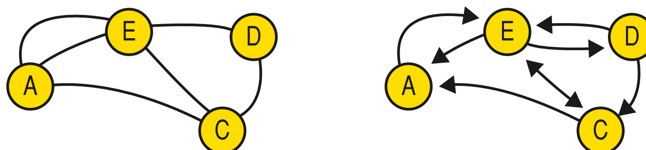
Ciclo



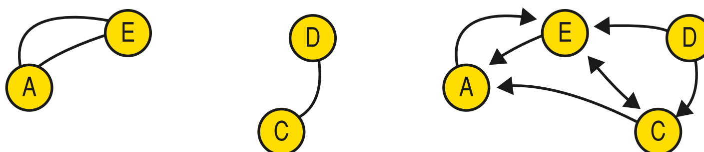
GRAFO CONNESSO E ACICLICO

Un grafo si dice connesso se ogni coppia di vertici (u, v) è collegata da un cammino semplice.
Un grafo si dice completamente connesso (o fortemente connesso) se ogni coppia di vertici (u, v) è collegata da un cammino orientato da u a v.
Un grafo si definisce aciclico se è privo di cicli.

In figura sono riportati esempi di grafi connessi e non connessi.

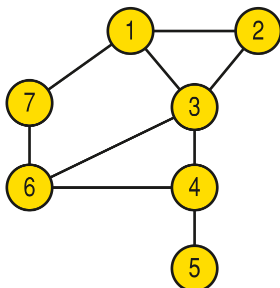


Connessi

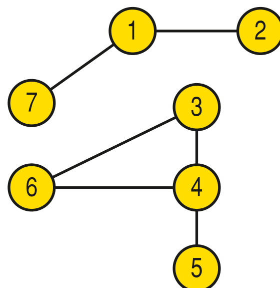


Non connessi

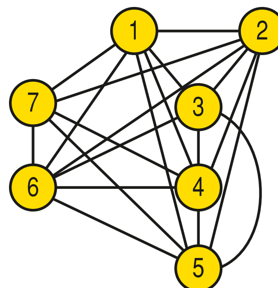
In figura sono riportati esempi di grafi con i diversi tipi di connessione.



Grafo connesso

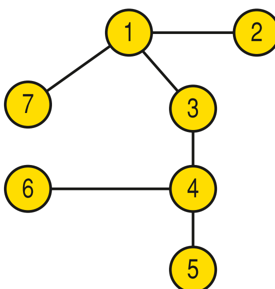


Grafo non connesso



Grafo completamente connesso

In figura è riportato un grafo aciclico.

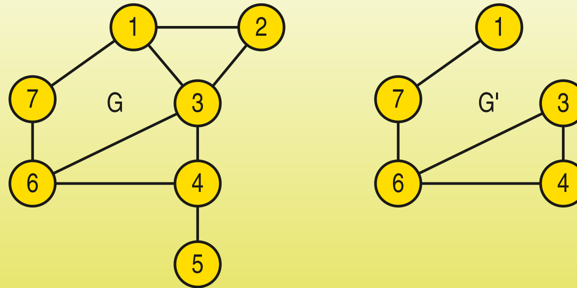


Un grafo non orientato, connesso e aciclico è un albero con n nodi e $n - 1$ archi.



SOTTOGRAFO

Un grafo G'=(V',E') è un sottografo di G=(V,E) se V' ⊆ V e E' ⊆ E.

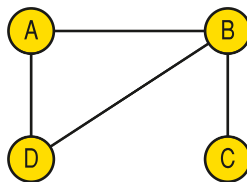


V = {1,2,3,4,5,6,7}
V' = {1,3,4,6,7}
E = {(1,2),(1,3),(1,7),(2,3),(3,4),(3,6),(4,5),(4,6),(6,7)}
E' = {(1,7),(3,4),(3,6),(4,6),(6,7)}

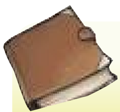
Rappresentazione dei grafi

Tra le possibili rappresentazioni dei grafi ricordiamo la matrice delle adiacenze: essendo la matrice una struttura con dimensionamento statico, questa rappresentazione si utilizza nel caso in cui il grafo si riferisca a situazioni dove il numero dei vertici rimane pressoché invariato.

Nella matrice delle adiacenze viene riportato, sia in ascissa sia in ordinata, l'elenco dei vertici: nelle celle di incrocio tra due nodi si indica con 1 la presenza di connessione e con 0 l'assenza. Osserviamo che se il grafo è non orientato la matrice presenta una simmetria diagonale.



	A	B	C	D
A	0	1	0	1
B	1	0	1	1
C	0	1	0	0
D	1	1	0	0

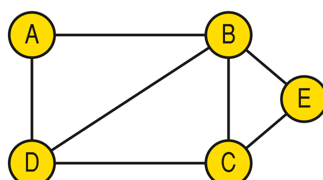


MATRICE DELLE ADIACENZE

Un grafo può essere rappresentato dalla sua matrice di adiacenza A, di dimensioni |V|x|V|, il cui generico elemento a_{ij} è definito nel modo seguente

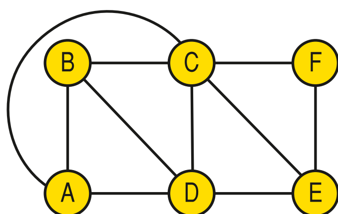
- a_{ij} = 1 se (i,j)>EE
- a_{ij} = 0 altrimenti

Vediamo un secondo esempio:



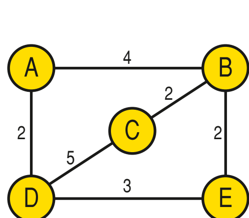
	A	B	C	D	E
A	0	1	0	1	0
B	1	0	1	1	1
C	0	1	0	1	1
D	1	1	1	0	0
E	0	1	1	0	0

Vediamo un terzo esempio:

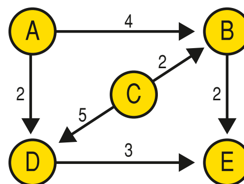


	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	1	1	0	0
C	1	1	0	1	1	1
D	1	1	1	0	1	0
E	0	0	1	1	0	1
F	0	0	1	0	1	0

L'utilizzo delle matrici si presta alla rappresentazione dei grafi pesati: basta sostituire nelle celle della matrice delle adiacenze al valore 1 il valore del peso tra il vertice rappresentato in ordinata e il vertice adiacente in ascissa, indicando con 0 o ∞ l'assenza di connessione.



	A	B	C	D	E
A	0	4	∞	2	∞
B	4	0	2	∞	2
C	∞	2	0	5	∞
D	2	∞	5	0	3
E	∞	2	∞	3	0

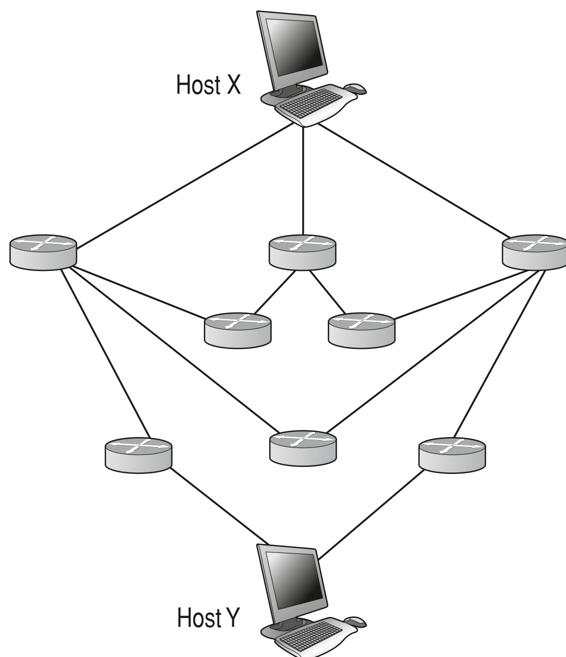


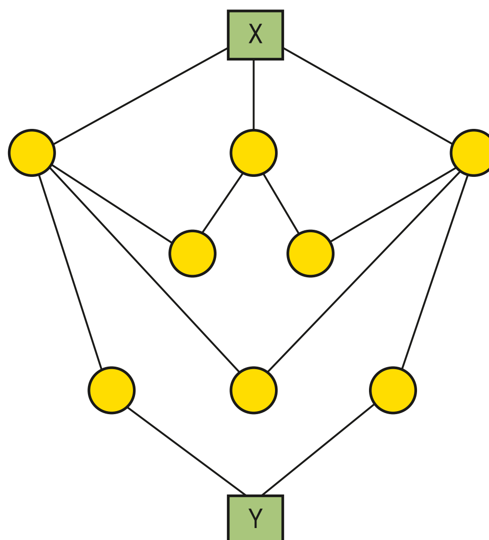
	A	B	C	D	E
A	0	4	∞	2	∞
B	∞	0	∞	∞	2
C	∞	2	0	5	∞
D	∞	∞	∞	0	3
E	∞	∞	∞	∞	0

Grafi e reti

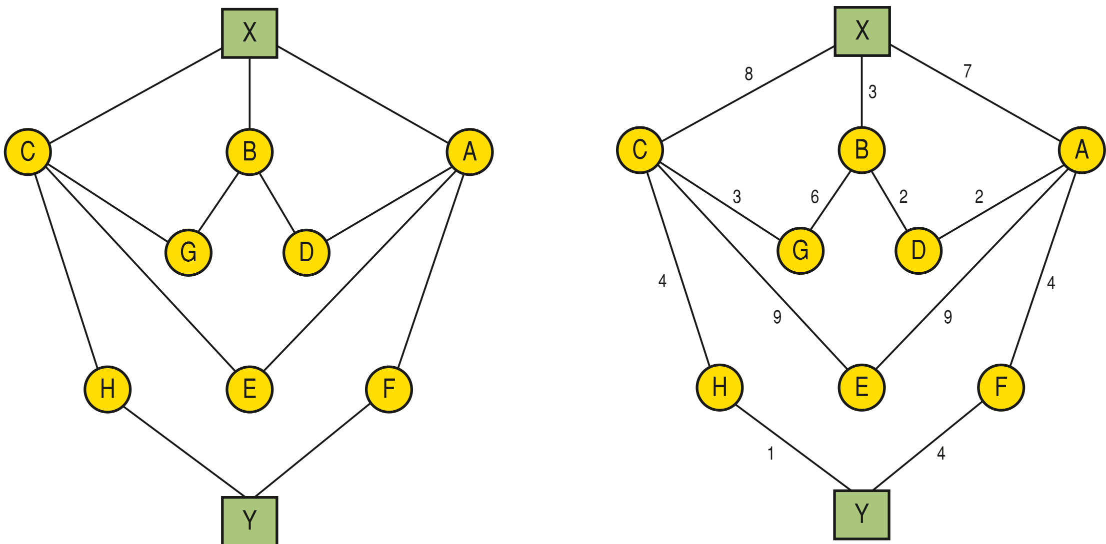
Vediamo ora come utilizzare i grafi per rappresentare le reti.

Supponiamo di dover trasmettere un messaggio dall'host X verso l'host Y che appartengono alla rete di figura:



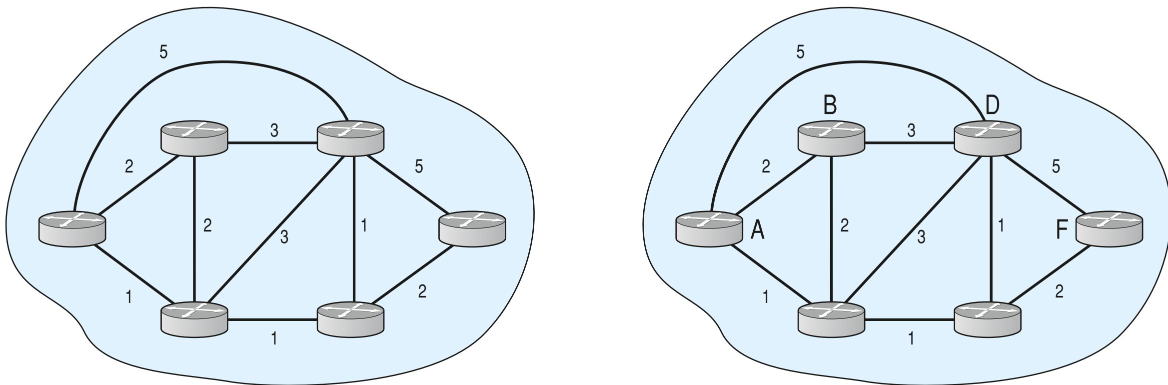


Come primo passaggio assegniamo un identificatore a ogni router e quindi successivamente li indichiamo come nodi e li colleghiamo tra loro come nella seguente figura.



A questo punto aggiungiamo i pesi agli archi e costruiamo la matrice delle adiacenze, dove il peso tra due router è il tempo (oppure il ritardo) per trasmettere un pacchetto tra di essi.

Vediamo un secondo esempio. Alla rete di figura assegniamo una etichetta a ogni router



e quindi la tabella delle adiacenze

	A	B	C	D	E	F
A	0	2	1	5	∞	∞
B	2	0	2	3	∞	∞
C	1	2	0	3	1	∞
D	5	3	3	0	1	5
E	∞	∞	1	1	0	2
F	∞	∞	∞	5	2	0

■ Ricerca del percorso minimo

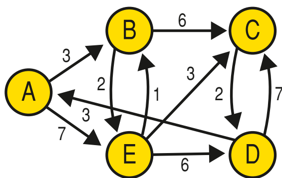
Sono molteplici gli algoritmi che effettuano la ricerca del percorso tra due nodi di un grafo così come gli algoritmi che ricercano i percorsi ottimi, cioè quelli con lunghezza minima (Shortest Path SP).

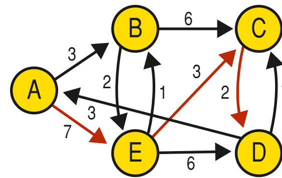


SHORTEST PATH

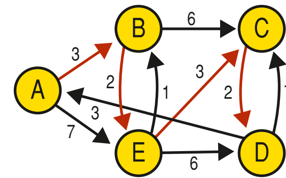
Uno shortest path (cammino minimo) dal nodo u al nodo v di V è un cammino p = (u,v1,v2,…,v) tale che w(p) è minimo.

Nel seguente esempio individuiamo su un grafo pesato orientato due cammini tra il nodo A e il nodo D.





p_1(A…D) = (A,E,C,D)
Peso di p_1= 7+3+2 = 12



p_2(A…D) = (A,B,E,C,D)
Peso di p_2= 3+2+3+2 = 10

In questo caso il cammino p_2 è il cammino minimo.

Per ogni destinazione è possibile costruire un percorso minimo e l'insieme dei percorsi minimi definisce quello che prende nome di albero d'inoltro (*forwarding tree*) o albero di consegna (*delivery tree*) dove il router sorgente assume il ruolo della radice e gli ultimi router di ogni percorso, quelli che hanno sul loro segmento gli host destinazione, prendono il nome di router foglia (*leaf router*).

L'algoritmo di **Dijkstra** permette di trovare i cammini minimi (◀ **Shortest Paths** ▶) in un grafo ciclico con pesi non negativi sugli archi: in particolare l'algoritmo può essere utilizzato *parzialmente* per trovare il cammino minimo che unisce due nodi del grafo, *totalmente* per trovare quelli che uniscono un nodo d'origine a tutti gli altri nodi e, ripetendolo più volte, per trovare tutti i cammini minimi da ogni nodo a ogni altro nodo.



◀ **Shortest paths** The shortest path is a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized. ▶

L'algoritmo di **Bellman-Ford** calcola i cammini minimi di un'unica sorgente su un grafo diretto pesato (dove alcuni pesi degli archi possono anche essere negativi).

Ma anche quando si è individuato il percorso ottimo per tutte le destinazioni il lavoro di un router non è finito: infatti la rete è per sua natura molto dinamica, dove nuovi router si inseriscono e altri vengono sostituiti inserendo modifiche nella topologia del grafo e nel peso dei suoi archi.
È quindi necessario che i nodi adiacenti siano in continuo dialogo, scambiandosi le informazioni che permettano l'aggiornamento della tabelle delle adiacenze e, di conseguenza, dei percorsi minimi di consegna.

Grafi, alberi e spanning tree ottimo

Un grafo può essere visto come un *caso particolare* di albero, e cioè:

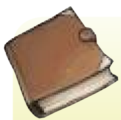


ALBERO

Un albero (tree) è un grafo non orientato, connesso e aciclico.

In un albero non possono esistere quindi percorsi chiusi (cicli) e per ogni coppia di nodi esiste uno e un solo cammino che li congiunge.

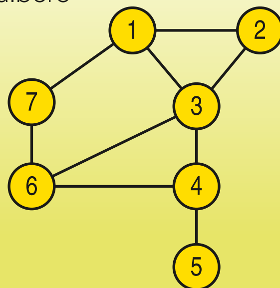
È possibile trasformare un grafo in un albero eliminando gli archi che determinano i cicli: in questo caso si ottiene un sottografo chiamato albero di ricoprimento (spanning tree).



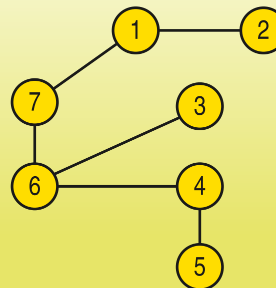
ALBERO DI RICOPRIMENTO

Dato un grafo connesso non orientato G=(V,E), uno ◀ **spanning tree** ▶ (albero di ricoprimento) di G è un sottografo G'=(V',T) tale che:

- G' è un albero
- V' = V



Grafo G

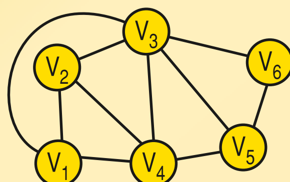


Spanning tree di G

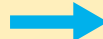


◀ **Spanning tree** A spanning tree *T* of a connected, undirected graph *G* is a tree composed of all the vertices and some (or perhaps all) of the edges of *G*. Informally, a spanning tree of *G* is a selection of edges of *G* that form a tree *spanning* every vertex. ▶

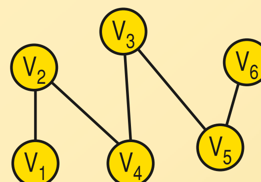
A partire da un grafo è possibile individuare molti alberi ricoprenti e all'aumentare del numero di cicli aumentano anche le differenti combinazioni:



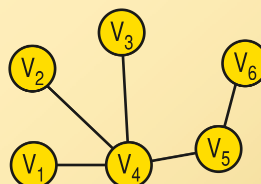
Grafo G







Spanning tree 1



Spanning tree 2

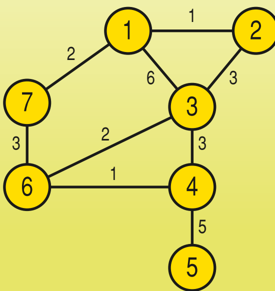
Tra tutti gli alberi ricoprenti individuiamo il Minimum Spanning Tree (MST) come l'albero avente somma degli archi minima.



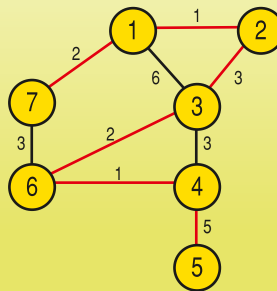
MINIMUM SPANNING TREE

Dato un grafo pesato connesso non orientato G=(V,E), un minimum spanning tree (albero di ricoprimento minimo) di G è uno spanning tree G'=(V,T) di peso minimo, cioè tale che risulti minimo w(T) dove:

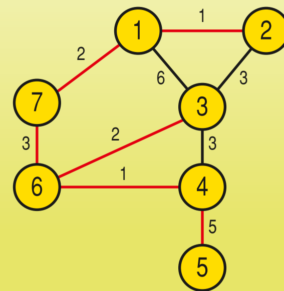
$$w(T) = \sum_{(i,j)\in T} w(i,j)$$



Grafo pesato G



Un MST di G
w(T) = 14



Un altro MST di G
w(T) = 14



Zoom su...

Tra gli algoritmi che effettuano la ricerca di uno Spanning Tree ricordiamo:
- algoritmo Breadth-First Search (BFS);

mentre quelli che ricercano un Minimum Spanning Tree:
- algoritmo di Kruskal;
- algoritmo di Prim.

Come è stato riportato nel disegno precedente per lo stesso grafo possono esistere diversi MST, cioè Spanning Tree aventi tutti lo stesso valore (minimo) come somma degli archi.



SPANNING TREE OTTIMO

Definiamo Spanning Tree Ottimo lo Spanning Tree formato dai cammini a costo minimo (shortest path) che uniscono il nodo radice agli altri nodi del grafo (prende anche il nome di ◀ **sink tree** ▶).



◀ **Sink tree** (minimum spanning tree): set of all optimal paths from all sources to a given destination. Can be found using the *shortest path* algorithm from destination to all sources giving optimal though not necessarily unique route. ▶

L'individuazione dei **sink tree** è l'obiettivo di un algoritmo di instradamento in un router e noi analizzeremo a tale scopo, nelle prossime lezioni, l'algoritmo di Dijkstra e di Bellman-Ford.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Il router di default dell'host mittente prende anche il nome di:
a) router origine
b) first router
c) di primo hop
d) router di partenza

2 In un grafo, effettuare la ricerca del cammino minimo tra due router equivale alla (indicare la risposta errata):
a) ricerca del cammino a minima distanza
b) ricerca del cammino con l'arco a minor peso
c) ricerca del cammino col minor numero di archi
d) ricerca del cammino a minima lunghezza

3 Un digrafo è:
a) un grafo orientato digitale
b) un grafo orientato diretto
c) un grafo orientato binario
d) un grafo non orientato digitale
e) un grafo non orientato diretto
f) un grafo non orientato binario

4 Nei seguenti cammini, indicare quale è semplice [S] e quale è una passeggiata [P]:
a) .... p_1=(1,2,3,1,7,6)
b) .... p_2=(2,3,1,7,6,2)
c) .... p_3=(11,2,3,1,7,6,5)
d) .... p_4=(11,2,3,1,7,6,5,2)
e) .... p_5=(1,2,3,6,5,4)

5 Nei seguenti cammini, indicare quale è un circuito [A] e quale è un ciclo [B]:
a) .... p_1=(1,2,3,1,7,1)
b) .... p_2=(2,3,1,7,6,2)
c) .... p_3=(5,2,3,1,7,6,5)
d) .... p_4=(2,5,3,1,4,1,7,5,2)
e) .... p_5=(1,2,3,6,5,4,1)

6 Un albero è:
a) un grafo non orientato, connesso e aciclico
b) un grafo non orientato, connesso e ciclico
c) un grafo orientato, connesso e aciclico
d) un grafo non orientato, connesso e ciclico
e) un grafo non orientato, non connesso e aciclico
f) un grafo non orientato, non connesso e ciclico

Test vero/falso

1 Il router predefinito è anche chiamato router di primo hop. V F
2 Un grafo non orientato è un grafo in cui gli archi sono coppie non ordinate di vertici. V F
3 Un grafo pesato è un grafo nel quale viene associata per ogni arco una funzione. V F
4 Con passeggiata si intende un cammino con nodi distinti. V F
5 Si definisce distanza la lunghezza del cammino più breve tra i vertici più distanti. V F
6 Il peso di un cammino è la somma dei pesi associati a tutti i lati che compongono il cammino. V F
7 Un grafo si definisce aciclico se è privo di cicli. V F
8 Uno shortest path dal nodo u al nodo v di V è un cammino tale che w(p) è minimo. V F
9 Se a un grafo si eliminano gli archi che determinano i cicli si ottiene uno spanning tree. V F
10 In un grafo esiste un solo Minimum Spanning Tree (MST) avente peso minimo. V F
11 L'individuazione dei sink tree si effettua con l'algoritmo di Dijkstra o di Bellman-Ford. V F

Verifichiamo le competenze

Esercizi

1 Definire la tabella delle adiacenze per una rete ad anello di 5 nodi, considerando un grafo orientato non pesato.

Dopo aver numerato i nodi da v1 a v5 una possibile topologia ad anello è rappresentata dalla seguente matrice delle adiacenze:

	V1	V2	V3	V4
V1				
V2				
V3				
V4				

2 Rappresentare graficamente la rete corrispondente alla seguente tabella delle adiacenze.

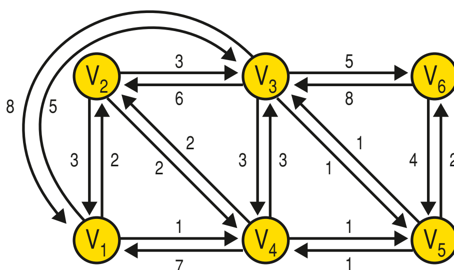
	V1	V2	V3	V4
V1		3	15	15
V2			0	0
V3				0
V4				

3 Una rete di 5 nodi è rappresentata dalla seguente tabella delle adiacenze.

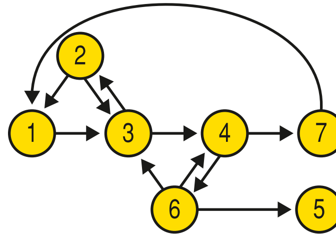
	V1	V2	V3	V4	V5
V1	0	10	5	0	0
V2	10	0	3	10	5
V3	5	3	0	0	0
V4	0	10	0	0	3
V5	0	5	0	3	0

Si disegni il grafo non orientato pesato corrispondente alla rete.

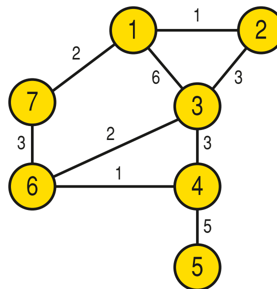
4 Dato il seguente grafo pesato costruire la matrice delle adiacenze.



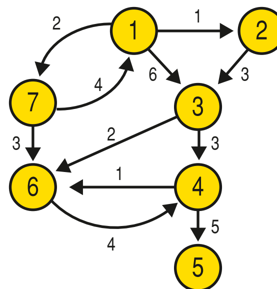
5 Dato il seguente grafo non pesato costruire la matrice delle adiacenze.



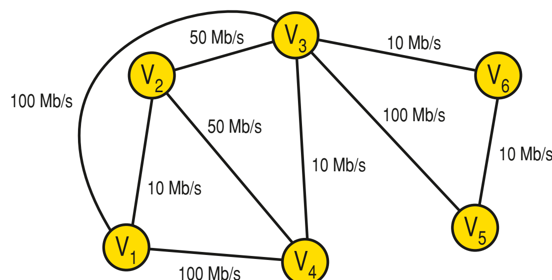
6 Data la rete rappresentata in figura si determini la tabella delle adiacenze considerando un grafo non orientato pesato. I collegamenti sono tutti bidirezionali e la loro capacità è indicata in figura.



7 Data la rete rappresentata in figura si determini la tabella delle adiacenze considerando un grafo orientato pesato. I collegamenti sono tutti bidirezionali e la loro capacità è indicata in figura.



8 Data la rete rappresentata in figura si determini la tabella delle adiacenze considerando un grafo orientato pesato. I collegamenti sono tutti bidirezionali e la loro capacità è indicata in figura. Si assegni come peso di ciascun collegamento il valore 100 diviso la capacità espressa in Mb/s.



Lezione 4

ALGORITMI DI ROUTING STATICI

IN QUESTA LEZIONE IMPAREREMO...

- le tipologie degli algoritmi statici
- a configurare manualmente una tabella di routing
- l'algoritmo di Dijkstra

Introduzione agli algoritmi statici

La classificazione degli algoritmi di routing nelle reti viene fatta sostanzialmente mediante due criteri:

A per come vengono gestite le informazioni sulla topologia:
- algoritmi distribuiti: in essi ciascun nodo conosce solo i propri vicini e i link che lo collegano ai propri vicini senza avere una visione globale della topologia di rete; i nodi attivano un processo interattivo scambiando tra nodi adiacenti le informazioni in loro possesso sulla metrica del segmento di rete del quale sono a conoscenza;
- algoritmi isolati: ogni LAN calcola in modo indipendente le tabelle di instradamento senza scambiare informazioni con le altre (come algoritmi di routing isolato ricordiamo hot potato e backward learning);
- algoritmi centralizzati: ciascun nodo ha conoscenza completa della topologia della rete e calcola i cammini migliori verso tutti gli altri nodi indipendentemente dal resto della rete.

B per adattabilità:
- algoritmi statici: sono algoritmi utilizzati nelle reti che cambiamo molto raramente in quanto non sono in grado di accorgersi dei mutamenti della struttura della rete stessa dato che i criteri di scelta dei percorsi sono decisi in anticipo, all'avvio della rete;
- algoritmi dinamici: sono in grado di ricostruire le tabelle non appena avviene un cambiamento nella rete (di traffico, di disposizione dei nodi ecc.) adattandosi quindi repentinamente alle nuove topografie dovute alla naturale evoluzione della rete e calcolando di conseguenza la soluzione migliore.

Noi analizzeremo gli algoritmi di routing in base alla adattabilità e in questa lezione affronteremo lo studio degli algoritmi statici.

Gli algoritmi di routing statici sono di tipo *non adattivo* (*static routing*) e le decisioni di routing sono prese in anticipo, all'avvio della rete, e comunicate ai router che poi vi si attengono sempre: possono inoltre essere classificati per tipo in tre gruppi:
- shortest path routing;
- flooding;
- flow-based routing.

Agli algoritmi presenti in queste categorie è necessario aggiungere il metodo "statico per eccellenza" utilizzato per la costruzione delle tabelle di routing, e cioè il metodo manuale.

Configurazione manuale delle tabelle di routing

Prima di analizzare gli algoritmi statici che permettono di definire le tabelle di routing è opportuno sottolineare che il metodo più diffuso per configurare i router è quello di fare manualmente l'inserimento dei dati nelle tabelle.

Spesso la configurazione manuale è anche la sola disponibile è viene fatta dall'amministratore di rete dopo aver effettuato l'analisi dei cammini possibili avendo a disposizione tutte le informazioni della rete, cioè topologia, caratteristiche dei link ecc.

Questo metodo risulta essere difficoltoso da utilizzarsi non appena aumenta le dimensione della rete e in presenza di topologie a grafo.

La presenza di grafi, infatti, rende complessa la determinazione "manuale" dei cammini ottimi e quindi difficilmente l'amministratore della rete riesce a definire tabelle che configurano la rete in modo ottimale.

La configurazione manuale viene comunque sempre utilizzata negli end-host nei quali è sempre prevista la presenza della route di default.
Per definire manualmente le tabelle di routing ci vengono in aiuto gli alberi di instradamento.

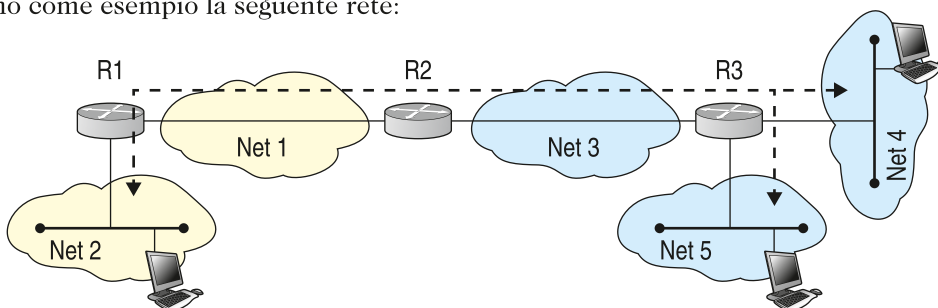
Albero di instradamento

La costruzione dell'albero di instradamento è molto semplice e viene effettuata a partire dallo schema della rete evidenziando il router mittente e "seguendo" il percorso che i pacchetti compiono per raggiungere tutte le destinazioni.

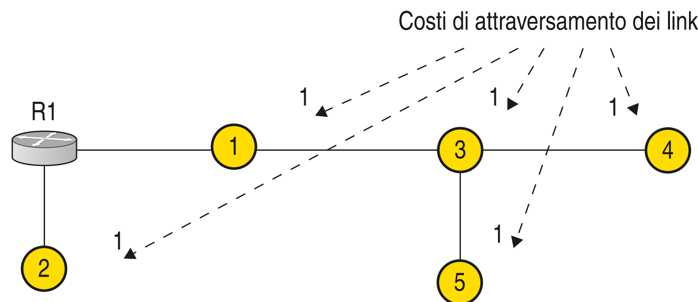
Come si è detto questo metodo è utilizzabile per reti di piccole dimensioni e con topologia semplice dove è immediato individuare il percorso più corto: vedremo che la ricerca dello shortest path verrà effettuata con appositi algoritmi nel caso di reti complesse.

ESEMPIO 5

Prendiamo come esempio la seguente rete:



dalla rete di figura si ottiene il seguente albero di instradamento



dove a ogni rete si sostituisce un nodo e a ogni router un arco.

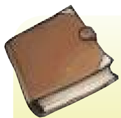
In questo caso il risultato si ottiene molto semplicemente perché oltre alle piccole dimensioni della rete non vi sono neppure cammini ciclici che porterebbero alla generazione di alternative nei percorsi.

Utilizzando gli alberi di instradamento è inoltre immediato determinare il numero di hop per raggiungere ogni destinazione e quindi completare la tabella di instradamento.

Principio di ottimalità

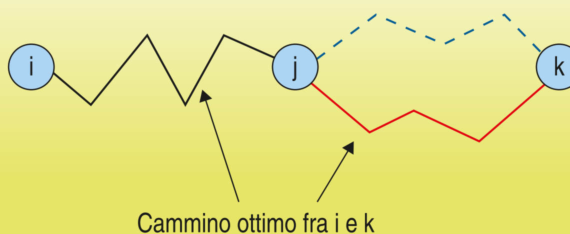
Nel caso di reti con cicli è necessario individuare il miglior percorso che collega ogni destinazione al router di partenza, che sappiamo essere il minimum spanning tree.

È possibile fare una considerazione generale sull'ottimalità dei cammini, chiamato principio di ottimalità:



PRINCIPIO DI OTTIMALITÀ

Il principio di ottimalità dice che se il router j è nel cammino ottimo fra i e k, allora anche il cammino ottimo fra j e k è sulla stessa strada:



Dimostriamolo per assurdo, ipotizzando cioè che nell'esempio di figura esista un cammino tra j e k (che nel disegno è tratteggiato in blu) con costo minore di quello indicato in rosso: se così fosse, ci sarebbe un altro cammino fra i e k migliore di quello ottimo (c.v.d.).

Una diretta conseguenza è che l'insieme dei cammini ottimi da tutti i router a uno specifico router di destinazione costituiscono un albero, detto sink tree per quel router: il compito degli algoritmi di routing è proprio quello di individuare i sink tree relativi a tutti i possibili router di destinazione in modo da instradare successivamente i pacchetti esclusivamente lungo tali sink tree.

Link State Packet

Gli algoritmi statici prendono anche il nome di algoritmi di link state in quanto ogni nodo dopo che ha acquisito le informazioni dei nodi a esso adiacenti e le relative distanze per raggiungerle, le invia a tutti gli altri nodi della rete mediante dei ◀ **Link State Packet (LSP)** ▶ contenenti i seguenti dati:

- stato di ogni link connesso al router;
- identità di ogni vicino connesso all'altro estremo del link;
- costo del link;
- numero di sequenza per l'LSP;
- checksum;
- lifetime.

Tale approccio è utilizzato nello standard ISO 10589 (protocollo IS-IS) e nel protocollo OSPF (adottato in reti TCP/IP).

Al ricevimento dei LSP ogni nodo si costruisce un database di LSP e una mappa completa della topologia della rete e sulla base di questa informazione provvede successivamente a calcolare i cammini minimi verso tutte le destinazioni.



◀ **Link State Packet LSP** is a packet of information generated by a network router in a link state routing protocol that lists the router's neighbors. ▶

Ogni LSP viene analizzato dal router e:

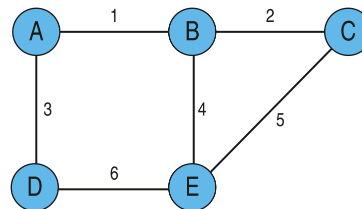
- se non ha mai ricevuto LSP da quel router o se l'LSP è più recente di quello presente nel proprio link state database lo memorizza e lo ritrasmette in ◀ flooding ▶ su tutte le linee eccetto quella da cui l'ha ricevuto;
- se l'LSP ha lo stesso numero di sequenza di quello posseduto non fa nulla;
- se l'LSP è più vecchio di quello posseduto trasmette al mittente il pacchetto più recente che è in suo possesso.



◀ Flooding Flooding significa allagamento. Infatti, per essere sicuri di aver bagnato una certa pianta, basta allagare tutto l'orto! ▶

ESEMPIO 6

La semplice rete riportata nella figura ▶



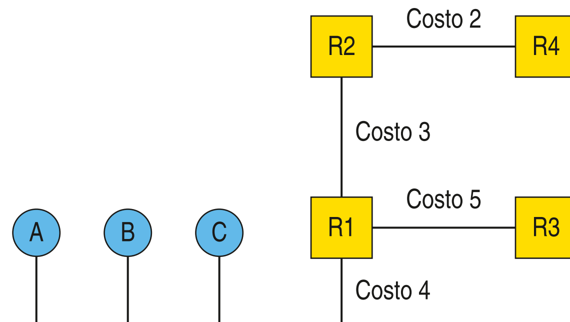
è rappresentata dal seguente database:

Da	Verso	Link	Costo	Numero di sequenza
A	B	1	1	1
A	D	3	1	1
B	A	1	1	1
B	C	2	1	1
B	E	4	1	1
C	B	2	1	1
C	E	5	1	1
D	A	3	1	1
D	E	6	1	1
E	B	4	1	1
E	C	5	1	1
E	D	6	1	1

Con questa tabella ogni nodo può calcolare il percorso più breve verso tutti gli altri nodi.

ESEMPIO 7

Vediamo ora un secondo semplice esempio di come viene realizzato un LS database.



Il router R1 ha direttamente collegati 6 host e di ciascuno conosce il peso del collegamento e costruisce la seguente tabella:

Adiacenze R1/x	Costo
A	4
B	4
C	4
R1	0
R2	3
R3	5

Quindi la invia in flooding su tutti i link del router e i dati vengono memorizzati nei diversi router formando una mappa completa e aggiornata della rete.

Analogamente il router 2 invia un messaggio così composto:

Adiacenze R2/x	Costo
R1	3
R2	0
R4	2

Lo stesso discorso vale per i router R3 e R4

Adiacenze R3/x	Costo
R1	5
R3	0

Adiacenze R4/x	Costo
R2	2
R4	0

Ricapitolando abbiamo:

R1	A/4	B/4	C/4	R2/3	R3/5
R2	R1/3	R4/2			
R3	R1/5				
R4	R2/2				

Dall'unione di tutte queste informazioni è possibile costruire la seguente tabella delle adiacenze:

	A	B	C	R1	R2	R3	R4
A	0	0	0	4			
B	0	0	2	4			2
C	0	2	0	4			
R1	4	4	4	0	3		
R2				3	0		
R3	5					0	
R4		2				5	0

Algoritmi statici: generalità

Shortest path routing

L'algoritmo di Dijkstra (che descriveremo in seguito) permette di calcolare il miglior percorso in un grafo in modo da ottenere per ogni router il sink tree verso ogni altro possibile nodo della rete: questo algoritmo viene mandato in esecuzione all'avvio della rete da un host di gestione che mantiene le informazioni di tutta la rete e invia i risultati a ogni router dove vengono memorizzati nel LSDB.



CRITERI DI MINIMIZZAZIONE

Sono diversi i criteri di minimizzazione dei percorsi: possiamo avere i percorsi migliori in base:
- alla lunghezza fisica dei collegamenti;
- al numero di archi da attraversare (numero di hop);
- al tempo medio di trasmissione;
- a una combinazione di lunghezza, banda trasmissiva, traffico medio ecc.

È la stessa possibile alternativa offerta dai navigatori satellitari, cioè quella di poter scegliere un itinerario cercando o di percorrere il numero minimo di chilometri oppure di impiegarci il minore tempo possibile.

Flooding

La tecnica del flooding consiste nell'inviare ogni pacchetto su tutte le porte di uscita (tranne naturalmente quella da cui è arrivato).

Un generico pacchetto verrà sicuramente ricevuto da tutti i nodi della rete compreso quello a cui è effettivamente destinato e quindi, in linea di principio, il flooding può essere usato come algoritmo di routing.

È facile constatare che con questa tecnica, dato che vengono generati un numero enorme di pacchetti che percorrono tutte le possibili strade, allo stesso nodo arrivano sicuramente molti pacchetti uguali.

È quindi necessario evitare che lo stesso pacchetto venga inoltrato all'infinito, a tal fine si introducono due semplici meccanismi di controllo:

A viene stabilito dal mittente un numero massimo di hop che il pacchetto può fare e viene inserito in un contatore che viene decrementato a ogni hop fino a giungere a 0;

B ogni router tiene memoria di ogni pacchetto che instrada, in modo tale che se lo riceve per la seconda volta lo ignora (ogni router per ogni pacchetto tiene una coppia di dati: *source router ID, sequence number*).

Il vantaggio della tecnica di flooding come *algoritmo di instradamento* è l'estrema semplicità dato che l'elaborazione eseguita in ciascun nodo è praticamente nulla: è particolarmente adatto alla trasmissione broadcasting dove l'informazione deve proprio essere inviata a tutti gli host.
Un altro campo nel quale viene utilizzato è quello militare, dove è necessario avere la massima sicurezza, massima affidabilità e robustezza anche a scapito dell'efficienza.

Questo algoritmo risulta essere il più semplice ed efficace in quanto trova sempre il percorso migliore nel minor tempo possibile ma, naturalmente, a scapito della efficienza dato che tende a saturare i canali trasmissivi: per questo motivo viene utilizzato come termine di paragone per gli altri algoritmi di ricerca del cammino ottimo.

Flow-based routing

Questo algoritmo calcola il percorso migliore in base al traffico medio di ogni linea, ipotizzandolo "abbastanza" costante nel tempo. I valori di ritardo per ogni linea vengono calcolati conoscendo le caratteristiche fisiche della linea stessa e la stima del traffico medio per ogni coppia di router.

Questo metodo è simile a quello che ogni automobilista utilizza per raggiungere una destinazione evitando le strade dove sa già, per esperienza, che saranno trafficate: se però tutti utilizzassero questo metodo di fatto nessuno passerebbe sulle strade trafficate, che diventerebbero deserte, e tutti gli automobilisti finirebbero "in coda" su strade alternative.

■ L'algoritmo di Dijkstra

L'algoritmo di Dijkstra è un algoritmo che rientra tra i shortest path routing e permette di calcolare l'albero dei cammini minimi tra un nodo di un grafo e tutti gli altri in modo da configurare le tabelle di routing: data la sua natura statica se cambia la configurazione della rete è necessario ricalcolare l'albero dei cammini minimi ripartendo da capo.
È un algoritmo di tipo centralizzato in quanto ciascun nodo calcola il cammino ottimo da se stesso verso tutti gli altri nodi in modo indipendente, partendo unicamente dalle informazioni complete sulla topologia della rete e sulle caratteristiche dei link.

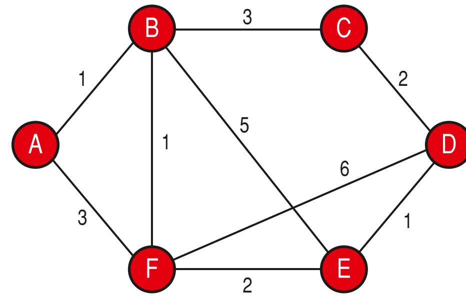
Prima di vedere come procede, definiamo innanzi tutto l'obiettivo che è quello di trovare i cammini minimi tra un nodo 1 (sorgente) e tutti gli altri nodi partendo dall'ipotesi che tutti gli archi abbiamo pesi positivi.

Il metodo di Dijkstra si assegna ai nodi delle etichette, che possono essere:
- temporanee: il costo massimo per la raggiungibilità del nodo in esame;
- permanenti: costo del cammino minimo.

L'algoritmo di calcolo modifica le etichette temporanee cercando di minimizzarle e di renderle permanenti.

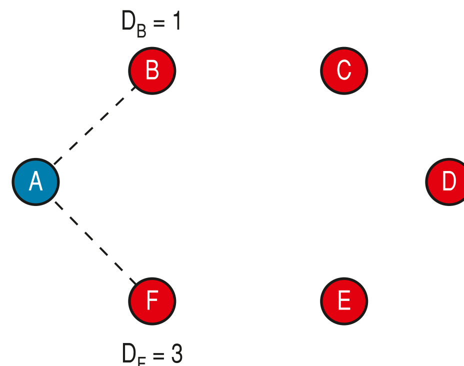
ESEMPIO 8

Vediamo un semplice esempio applicando l'algoritmo al grafo non orientato della figura: ▶

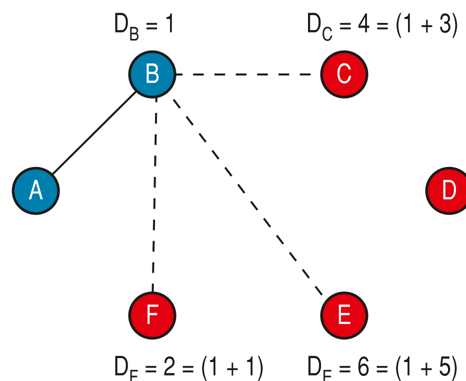


Scegliamo "a piacere" un nodo, in quanto il risultato che otteniamo è ininfluente dal nodo di partenza, e iniziamo a scrivere le etichette provvisorie sui due nodi a esso adiacenti.

Partiamo da A ed etichettiamo B con 1 e F con 3. ▶

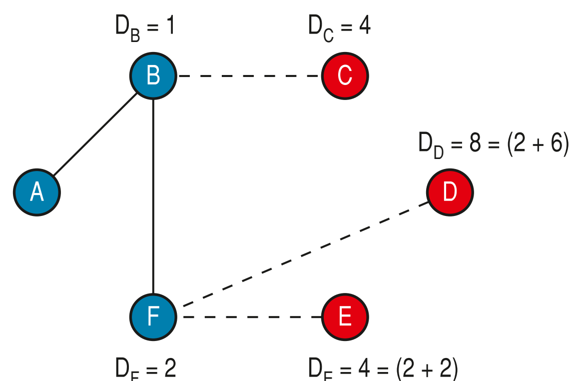


Scegliamo ora il "percorso meno costoso" che si è individuato, cioè quello che parte dal nodo B, e ripetiamo le stesse operazioni assegnando le etichette provvisorie ai nodi C, F ed E ottenute come somma dei percorsi dal nodo precedente (1) e dei nuovi pesi dei rispettivi archi: ▶



La scelta secondo la quale il nodo successivo da visitare è quello più vicino a uno dei nodi già visitati dà il nome all'algoritmo Shortest Path First, cioè prima il percorso più breve.

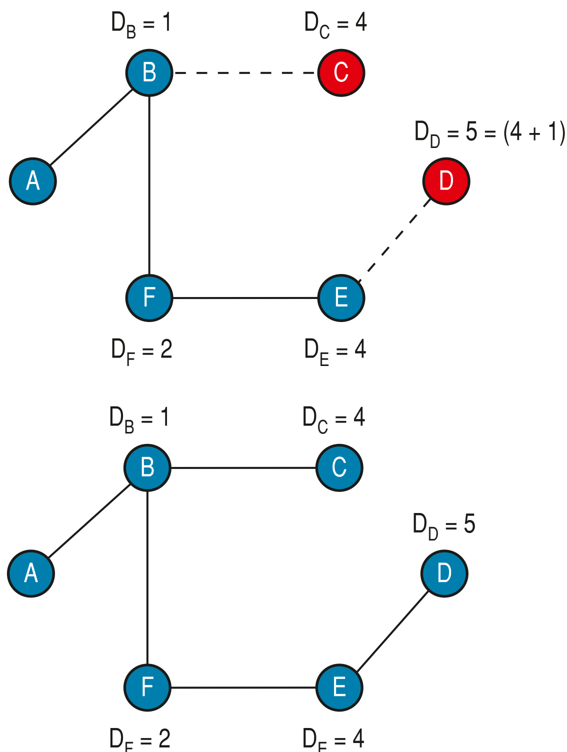
Procediamo in modo analogo scegliendo il percorso più corto tra i 3 sino a ora individuati, cioè quello che ha raggiunto il nodo F (peso 2), e aggiorniamo le etichette dei nodi a esso adiacenti ($D_D = 2 + 6 = 8$, $D_E = 2 + 2$): ▶



Con lo stesso criterio il prossimo nodo da analizzare è il nodo E (peso 4): dal nodo E possiamo raggiungere il nodo D con peso $D_D=4+1=5$ che è minore del precedente valore dell'etichetta provvisoria: la sostituiamo con una *nuova etichetta provvisoria*. ▶

Il nodo con peso minore tra quelli che non sono ancora sati analizzati è il C che ha peso 4, ma non porta migliorie in quanto la sua distanza da D ha peso 2 che peggiorerebbe l'etichetta parziale.

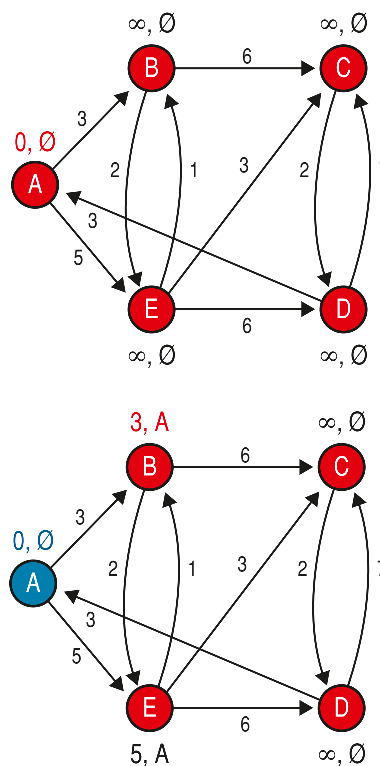
Ultimo nodo è il nodo D, ma anch'esso non introduce migliorie e quindi le etichette provvisorie diventano permanenti e si ottiene il seguente albero minimo: ▶



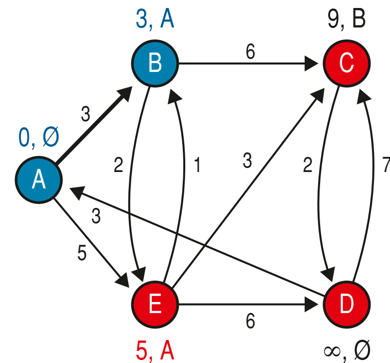
ESEMPIO 9

Vediamo ora un secondo esempio in un grafo orientato dove all'etichetta provvisoria che indica il peso aggiungiamo un secondo dato che indica il nodo antecedente che lo ha raggiunto: all'inizio le etichette vengono poste a ∞ e il nodo di provenienza viene indicato con il valore 0. ▶

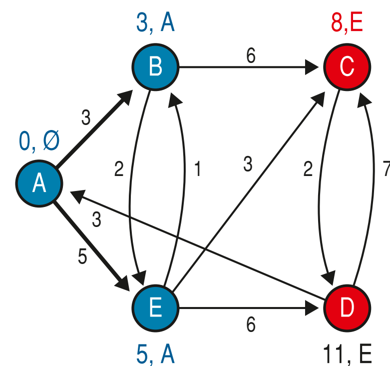
Partendo da A etichettiamo i nodi B(3,A) e E(5,A). ▶



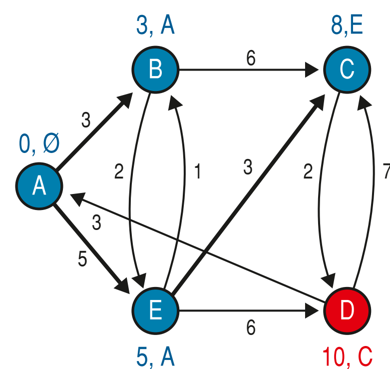
Il nodo successivo è B che è adiacente a C(3+6,B) e a E(3+2, ?), che è già etichettato, e il nuovo peso 5 non migliora quello provvisorio che ha nella sua etichetta, e quindi lo lasciamo invariato. ▶



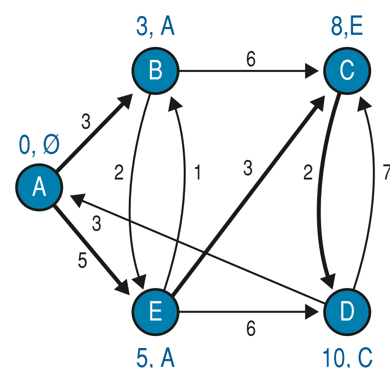
Il nuovo percorso minimo è quello che raggiunge il nodo E, quindi analizziamo i suoi nodi adiacenti C(5+3,E), che migliora la precedente etichetta e quindi viene aggiornata, e D(5+6,E) che è la prima etichetta provvisoria per il nodo D. ▶



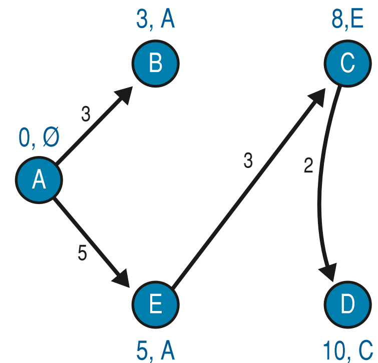
Tra i nodi D e C il Shortest Path è quello che raggiunge il nodo C, e attraverso esso si migliora la raggiungibilità del nodo con D(8+2,C). ▶



Ultimo a essere analizzato è il nodo D che non introduce migliorie, quindi il risultato è il seguente: ▶



e cancellando gli archi "non utilizzati" otteniamo il seguente albero minimo ▶





Zoom su...

PSEUDOCODIFICA

Vediamo un esempio di codifica in linguaggio di progetto dell'algoritmo di Dijkstra, dopo aver introdotto la notazione che verrà utilizzata per comodità nello pseudocodice:

- c(x,y): costo del collegamento dal nodo x al nodo y; è posto a ∞ se non sono adiacenti;
- D(v): costo del cammino dal nodo origine alla destinazione *v* per quanto riguarda l'iterazione corrente;
- p(v): immediato predecessore di *v* lungo il cammino;
- N': sottoinsieme di nodi per cui il cammino a costo minimo dall'origine è definitivamente noto;
- N: insieme di tutti i nodi presenti nel grafo.

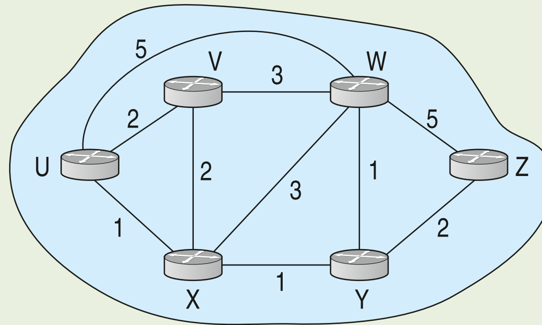
```
 1 Inizializzazione
 2 N' = {u}
 3 per tutti i nodi v // oppure finché N' = N
 4  se v è adiacente a u
 5    allora D(v) = c(u,v)
 6    altrimenti D(v) = ∞
 7   ripeti
 8    determina un w non in N' tale che D(w) sia minimo
 9   aggiungi w a N'
10   per ciascun nodo v adiacente a w e non in N'
11     aggiorna D(v) = min(D(v), D(w) + c(w,v))
```

Dove con l'istruzione

```
11 D(v) = min(D(v), D(w) + c(w,v))
```

si assegna a D(v) il nuovo costo verso v che è il valore minimo tra il vecchio costo presente nella etichetta temporanea oppure il costo del cammino minimo noto verso w più il costo da w a v.

Lo applichiamo alla seguente rete ▶



ottenendo: ▼

Passo	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	**1,u**	∞	∞
1	ux	2,u	4,x		**2,x**	∞
2	uxy	**2,u**	3,y			4,y
3	uxyv		**3,y**			4,y
4	uxyvw					**4,y**
5	uxyvwz					

Conclusioni

L'algoritmo di Dijkstra individua il cammino a lunghezza minima tra un nodo e tutti gli altri nodi di un grafo G procedendo per passi successivi e terminando quando si sono esplorati tutti i nodi e si è ottenuto uno spanning tree contenente i cammini a costo minimo tra il nodo sorgente e tutti gli altri nodi del grafo.

È doveroso ricordare che per poter effettuare tali calcoli è necessario avere a disposizione tutte le informazioni sulla topologia della rete.

La complessità di calcolo di questo algoritmo è $O(n^2)$, così ottenuta dal calcolo del numero di etichette da calcolare: $(n - 1) + (n - 2) + (n - 3) + \ldots + 2 + 1 = n(n - 2)/2$.
Esistono anche implementazioni che, grazie a ottimizzazioni, hanno complessità $O(n \cdot \log n)$.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 La classificazione degli algoritmi di routing in base alla topologia prevede:

a) algoritmi distribuiti, isolati, centralizzati
b) algoritmi locali, remoti, isolati
c) algoritmi distribuiti, isolati, locali
d) algoritmi remoti, isolati, centralizzati

2 Gli algoritmi di routing statici possono inoltre essere classificati per tipo in tre gruppi:

a) shortest path routing
b) flooding
c) link state
d) flow-based routing

3 Il routing statico:

a) prevede che il router sia preconfigurato
b) prevede che l'amministratore configuri il router
c) prevede che la rete non subisca variazioni per l'intera durata della configurazione del router
d) prevede che la rete non subisca variazioni per l'intera vita della rete

4 In che cosa consiste l'hot potato?

a) Un protocollo utilizzato dai bridge
b) Un arbitraggio delle reti locali
c) Un algoritmo di routing isolato
d) Un algoritmo di routing dinamico

5 Un Link State Packet contiene i seguenti dati (indicare quelli non presenti):

a) stato di ogni link connesso al router
b) identità di ogni vicino connesso all'altro estremo del link
c) costo del link
d) numero di host adicenti
e) numero di sequenza per l'LSP
f) numero di pacchetti nella sequenza
g) checksum
h) lifetime

6 Tra i criteri di minimizzazione dei percorsi possiamo utilizzare:

a) la lunghezza fisica dei collegamenti
b) il tempo medio di trasmissione
c) una combinazione di numero di messaggi, banda trasmissiva, dimensione del messaggio
d) il numero di hop

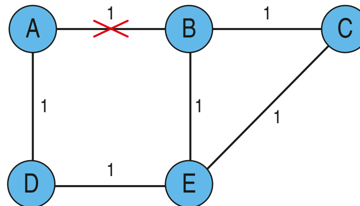
Test vero/falso

1 Gli algoritmi di routing statici sono di tipo adattivo. V F
2 Tra gli algoritmi di routing statici è incluso il metodo manuale. V F
3 La configurazione manuale viene comunque sempre utilizzata negli end-host. V F
4 Con sink tree si intende l'insieme dei cammini ottimi da tutti i router a uno specifico router. V F
5 La tecnica del flooding consiste nell'inviare ogni pacchetto su tutte le porte di uscita. V F
6 La tecnica del flooding è particolarmente adatta alla trasmissione broadcasting. V F
7 L'algoritmo di calcolo modifica le etichette temporanee per minimizzarle e renderle permanenti. V F
8 Il metodo di Dijkstra si utilizza solo sui grafi non orientati. V F

Verifichiamo le competenze

Esercizi

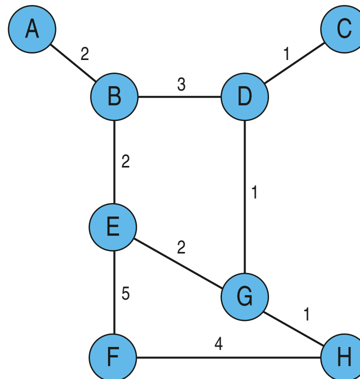
1 Supponiamo che nella rete riportata nella figura seguente si interrompa un collegamento, per esempio quello tra il nodo A e il nodo B.



Come viene modificato il database?

Da	Verso	Link	Costo	Numero di sequenza
A	B	1		
A	D	3		
B	A	1		
B	C	2	1	1
B	E	4	1	1

2 Dal seguente grafo definisci il link state database e la tabella delle adiacenze.



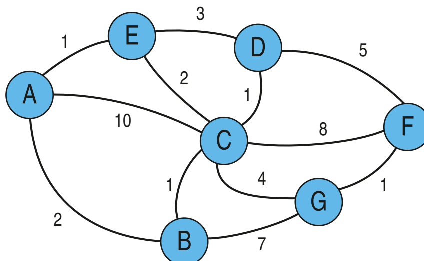
Il database è il seguente:

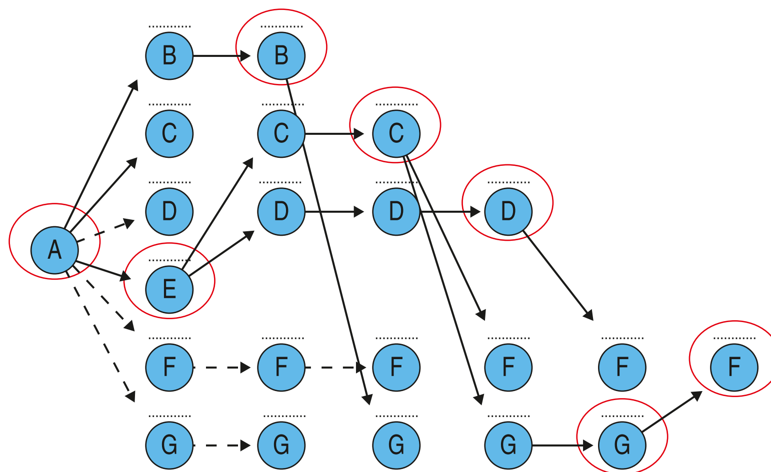
A	B/2		
B	A/3	D/3	E/2
C			
D			
E			
F			
G			
H			

che può essere semplicemente trasformata in una tabella delle adiacenze:

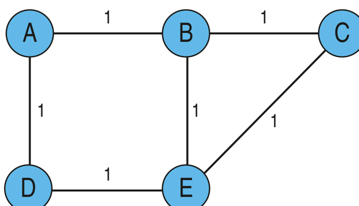
	A	B	C	D	E	F	G	H
A	0							
B		0						
C			0					
D				0				
E					0			
F						0		
G							0	
H								0

3 Al grafo di figura applica l'algoritmo di Dijsktra per ottenere il Minimum Spanning Tree.



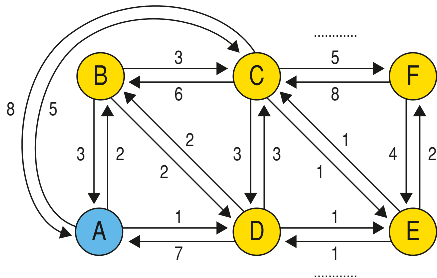


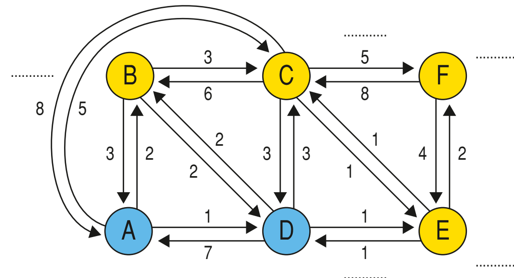
4 Al grafo di figura applica l'algoritmo di Dijsktra per ottenere la tabella di routing del nodo A.

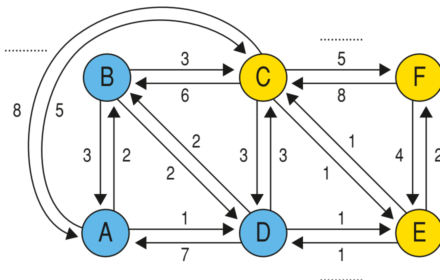


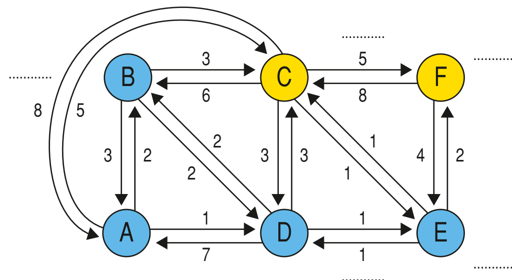
Da A verso	Link	Costo
A		
B		
D		
C		
E		

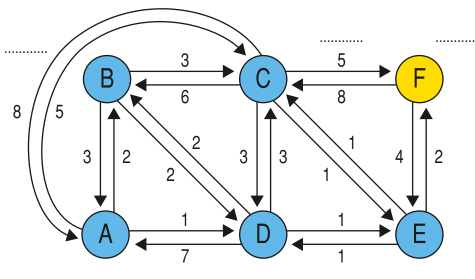
5 Al grafo di figura applica l'algoritmo di Dijsktra etichettando a ogni iterazione i rispettivi nodi.

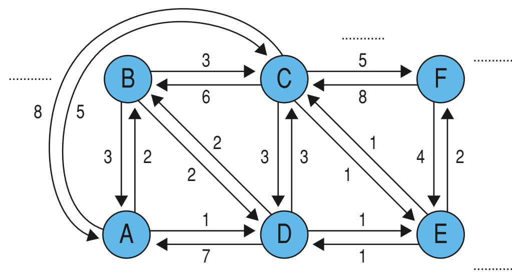




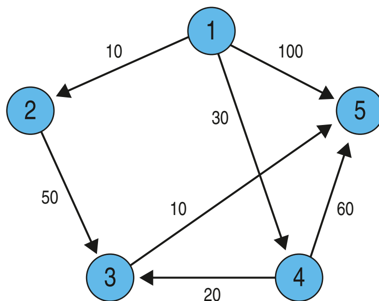






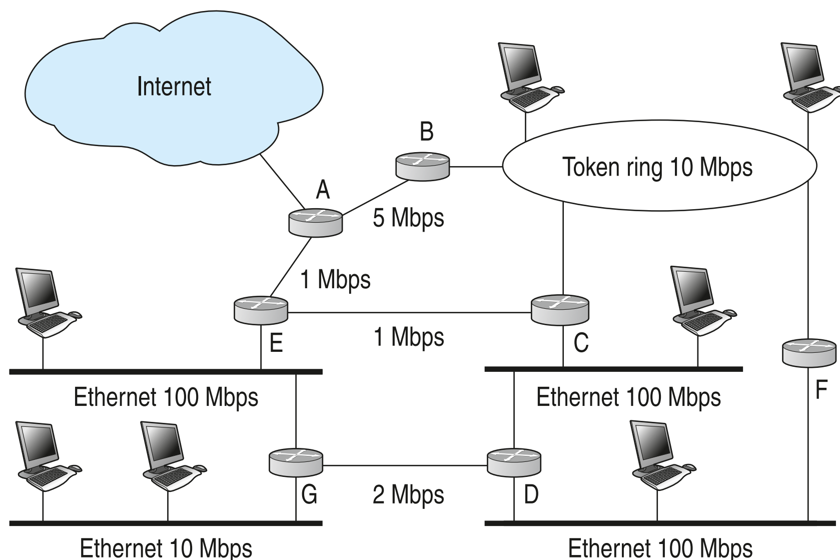


6 Individua l'albero dei cammini minimo del seguente grafo indicando i singoli passi dell'algoritmo di Dijkstra.



S	2	3	4	5

7 Nella rete in figura i router sono indicati con lettere maiuscole.



a) Applica l'algoritmo di Dijkstra per trovare il percorso a costo minimo dal router A a tutti gli altri router, considerando peso unitario su tutti i link.

b) Applica l'algoritmo di Dijkstra per trovare il percorso a costo minimo dal router A a tutti gli altri router, considerando un peso per link pari a 100/c, dove c è il bitrate del link in Mbps indicato in figura.

Lezione 5

ALGORITMI DI ROUTING DINAMICI

IN QUESTA LEZIONE IMPAREREMO...

- le caratteristiche degli algoritmi dinamici di routing
- l'algoritmo di Bellman-Ford

Introduzione agli algoritmi dinamici

Anche se gli algoritmi statici trovano il percorso ottimo, il risultato è relativo a una particolare situazione della rete in uno specifico istante di tempo e sono quindi adatti solo per le reti che cambiamo molto raramente dato che non sono in grado di accorgersi dei mutamenti della struttura della rete stessa.

Non possono essere utilizzati da soli per le reti geografiche in quanto queste, proprio per loro natura, sono strutture dinamiche in continua mutazione e continuano sia a crescere di dimensione che a modificare e sostituire dispositivi, dato che i componenti hardware sono soggetti a usura e obsolescenza tecnologica e vengono cambiati con prodotti nuovi più performanti.

Agli algoritmi statici vengono affiancati degli algoritmi dinamici che si occupano di scoprire i percorsi ottimi su una rete e di mantenerli aggiornati al variare dello stato della rete stessa, adattandosi quindi repentinamente alle nuove topografie della rete e calcolando di conseguenza la nuova soluzione migliore.

Tra gli algoritmi dinamici in questa lezione verrà descritto l'algoritmo di Bellman-Ford, comunemente chiamato distance vector routing, che è l'algoritmo oggi più utilizzato in Internet.

Algoritmo di Bellman-Ford

L'algoritmo di Bellman-Ford è un algoritmo distribuito che calcola per ciascun nodo il cammino migliore (e quindi definisce la sua tabella di routing) verso tutti gli host presenti sulla rete in base a informazioni che riceve dagli altri nodi, senza avere la conoscenza della topologia o delle caratteristiche dei link della rete: un nodo conosce solo gli host nelle sue "immediate vicinanze" (◀ neighbour ▶), cioè quelli che sono direttamente connessi con lui, e comunica solo con essi.



◀ Neighbour Letteralmente "vicini di casa", con neighbour si intendono gli host confinanti, direttamente raggiungili con un unico hop. ▶

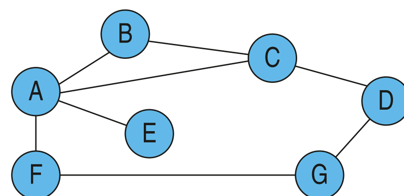
È grazie alla continua comunicazione tra nodi adiacenti, che si scambiano informazioni aggiornate sulle distanze dei proprio vicini, che l'algoritmo di Bellman-Ford ricalcola periodicamente i cammini ottimi rimanendo sempre aggiornato sulla nuova configurazione della rete.

Quindi ogni nodo ricalcola le proprie tabelle e le comunica agli altri nodi che, di seguito, rieffettuano i conteggi… e così via, si innesca un processo interattivo che porta alla convergenza per tutta la rete verso la soluzione ottima.

Per come opera, questo algoritmo richiede solamente l'operazione di scambio di informazioni tra vicini, e il conseguente calcolo della tabella di routing avviene nello stesso tempo su tutti i nodi ma in modo autonomo: è quindi un sistema di tipo asincrono.

L'algoritmo di Bellman-Ford è un algoritmo distribuito, iterativo e asincrono.

Descriviamolo mediante un semplice esempio: una rete è rappresentata dal grafo a fianco dove ogni nodo conosce i costi dei collegamenti ai nodi adiacenti che per semplicità ipotizziamo essere di costo unitario. ▶



Alla prima accensione indichiamo nella tabella costo 1 nei collegamenti diretti mentre per i nodi non adiacenti dei quali ancora non sappiamo nulla (o i collegamenti interrotti) registriamo un costo infinito: la situazione iniziale è quella riportata di seguito dove ogni nodo conosce solamente i suoi vicini e tramite questa conoscenza si costruisce la seguente tabella delle adiacenze:

	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

Da essa è possibile ricavare il percorso minimo e quindi la tabella di instradamento, che per il router B alla prima accensione è la seguente:

Destinazione	Next hop	Costo
A	A	1
C	C	1

e contiene cioè solamente le due righe inserite staticamente che riportano i due nodi a esso adiacenti.

I router iniziano a comunicare tra loro e ogni nodo spedisce aggiornamenti ai suoi nodi adiacenti e, quindi, riceve aggiornamenti dai suoi vicini: la trasmissione dei dati tra nodi può avvenire o periodicamente oppure ogni volta che in un nodo viene modifica la propria tabella (◀ **triggered update** ▶).



◀ **Triggered update** A triggered update is a routing table update that is sent immediately in response to a routing change. Triggered updates do not wait for update timers to expire. The detecting router immediately sends an update message to adjacent routers. ▶

I collegamenti che avevano all'origine peso infinito ora possono essere sostituiti dai valori ottenuti/calcolati, e dopo un breve periodo la tabella della adiacenze converge a questi dati:

	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Vediamo come viene costruita: come aggiornamento un router riceve dai suoi vicini un vettore distanza composto dalla coppia (destinazione, costo) presente nella tabella di instradamento.
Quando un nodo riceve dal suo vicino un aggiornamento analizza la tabella e se la nuova indicazione segnala un cammino migliore rispetto a quello che lui ha memorizzato, lo sostituisce con questo nuovo valore.

Per esempio, se il nodo A invia a B un messaggio nel quale gli segnala che E viene raggiunto con costo 1, B analizza la sua tabella e trova il valore ∞: dato che la nuova segnalazione è migliore di quella esistente sostituisce nella tabella in corrispondenza del nodo E il valore 2 (ottenuto dalla somma della distanza tra A ed E e tra B ed E).

Il router B aggiornerà anche la routing table, aggiungendo (o modificando) la riga corrispondente: nel nostro esempio viene aggiunta la terza riga

Destinazione	Next hop	Costo
A	A	1
C	C	1
E	A	2

Osserviamo che come next hop viene scritto il nome del nodo che ha proposto l'aggiornamento del dato.

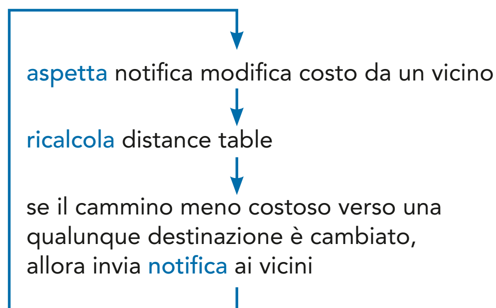
Il router B comunica questo nuovo valore ai nodi che gli sono adiacenti che, a loro volta, aggiornano le loro tabelle e inviano gli aggiornamenti che, a "cascata", raggiungono tutta la rete.

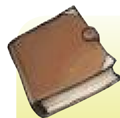
Dopo aver scambiato diversi aggiornamenti con i nodi adiacenti, tutti i nodi conosceranno i cammini migliori verso tutti gli altri nodi e completeranno la tabella di routing, che per il nodo B è la seguente.

Destinazione	Next hop	Costo
A	A	1
C	C	1
D	C	2
E	A	2
F	A	2
G	A	3

L’algortimo

Un affinamento dell’algoritmo che viene eseguito su ciascun host è il seguente:





FORMULA DI BELLMAN-FORD

Nell’algoritmo viene utilizzata quella che prende il nome di formula di Bellman-Ford che definisce il costo del percorso a costo minimo dal nodo *x* al nodo *y* (distanza minima), indicandolo con $d_x(y)$:

$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

Il significato della formula è il seguente: “il costo di andare da x a y *è minimo* se passo attraverso un nodo v tale che il costo totale di andare da x a v e poi da v a y è minimo”:

- x sa quanto ci mette per arrivare fino a v, indicato con c(x,v);
- v dice a x quanto lui impiega per arrivare fino a y, indicato con $d_v(y)$.

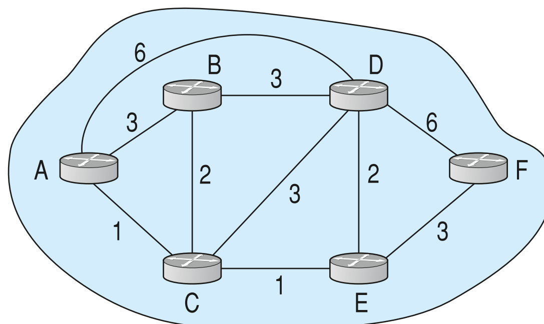
In altre parole “afferma una verità “ cioè che il cammino a costo minimo da x a y non può che essere la prosecuzione del cammino a costo minimo da uno dei nodi v vicini di x al nodo y.

La formula è ricorsiva, in quanto il secondo addendo è a sua volta una distanza ottenuta della formula di Bellman-Ford.

ESEMPIO 10

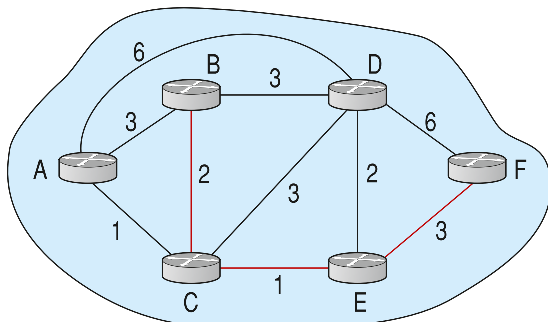
Vediamo un esempio applicando la formula sopra descritta al grafo di figura per individuare il percorso minore tra A e F conoscendo le distanze minime tra i nodi B, C e D adiacenti di A:

- distanza minima tra B e F: $d_B(F) = 6$
- distanza minima tra C e F: $d_C(F) = 4$
- distanza minima tra D e F: $d_D(F) = 5$

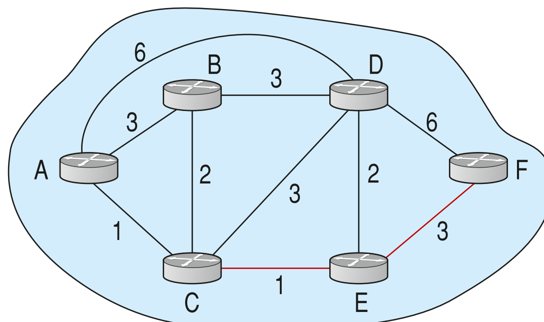


Infatti

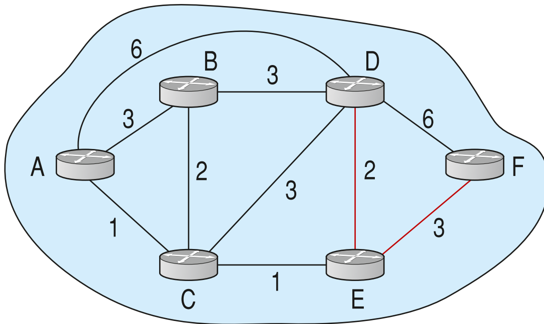
$d_B(F) = 6$



$d_C(F) = 4$



$d_D(F) = 5$



Nella precedente lezione abbiamo applicato l'algoritmo di Djikstra alla stesso grafo ottenendo, naturalmente, lo stesso risultato.

Le distanze tra il nodo A e i suoi adiacenti sono rispettivamente c(A,B) = 3, c(A,C) = 1, c(A,D) = 6.

Applichiamo la formula di Bellman-Ford (BF) e otteniamo:

$$d_A(F) = \min \{c(A,B) + d_B(F), c(A,C) + d_C(F), c(A,D) + d_D(F) \} = \min \{3 + 6, 1 + 4, 6 + 5\} = 5$$

Quindi A instrada a F tramite C con costo 5.

ESEMPIO 11

Vediamo un secondo esempio dove eseguiamo passo passo l'algoritmo introducendo la seguente notazione:

- indichiamo la tabella di distanze in questo modo:

		Costo via		
	D^A	B	C	E
Destinazione	B			
	C			
	D			
	E			
	F			

dove con D^A si indica che le distanze sono riferite al nodo A:
– in ordinata sono riportate le destinazioni;
– in ascissa il costo riferito alla via utilizzata (nodi vicini ad A).

- indichiamo con $d_x(y)$ la stima del costo del percorso a costo minimo da x al nodo y.

Inoltre ipotizziamo che:

- il nodo x conosce il costo del percorso verso ciascuno dei suoi vicini v: c(x,v);
- il nodo x mantiene in memoria un vettore di distanze D_x all'interno del quale ci sono le distanze stimate verso tutte le destinazione note: indichiamo il Distance Vector con $D_x = [d_x(y): y \in N]$;
- il nodo x mantiene in memoria anche il Distance Vector dei suoi vicini, cioè i $D_v = [d_v(y): y \in N]$ dove con v indichiamo i nodi adiacenti a x.

La tabella delle distanze completa per ogni nodo viene spesso indicata come una tabella a doppia entrata che in ordinata riporta il DV degli host adiacenti e in ascissa gli host di destinazione, come indicato di seguito:

		Costo verso		
		x	y	z
Da	x			
	y			
	z			

L'idea che sta alla base dell'algoritmo è che ogni nodo invia una copia del proprio distance vector a ciascuno dei suoi vicini che lo salva e, con i nuovi valori ricevuti, applica la formula BF per aggiornare in proprio distance vector come segue:

$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$ per ciascun nodo y in N

Finché tutti i nodi continuano a cambiare i propri DV in maniera asincrona, ciascuna stima dei costi viene modificata e converge a $d_x(y)$.

La rete che analizziamo è riportata nella figura a fianco ▶ dove iniziamo l'analisi ipotizzando che ogni router conosca i propri vicini e i pesi degli archi che lo collegano con essi che riportiamo nelle seguenti tabelle all'istante iniziale (cold starts).

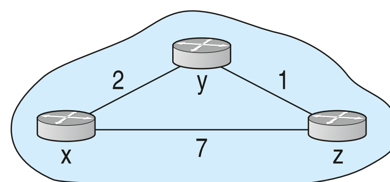


Tabella del nodo x

		Costo verso		
		x	y	z
Da	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		Costo via	
	D^X	y	z
Dest.	y	(2)	∞
	z	∞	(7)

Tabella del nodo y

		Costo verso		
		x	y	z
Da	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		Costo via	
	D^Y	x	z
Dest.	x	(2)	∞
	z	∞	(1)

Tabella del nodo z

		Costo verso		
		x	y	z
Da	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		Costo via	
	D^Z	x	y
Dest.	x	(7)	∞
	y	∞	(1)

Ogni router manda la propria tabella ai propri neighbour, che applicano anch'essi la formula BF per effettuare l'aggiornamento.

Vediamo cosa avviene in ciascuno di essi quando gli giunge l'aggiornamento dai vicini, a partire dal router x.

Nodo x

- Per prima cosa aggiorna le righe della tabella dove è presente ∞, che indica di non avere informazioni;
- in base ai nuovi valori ricalcola le distanze verso gli altri nodi ottenute come somma di percorsi parziali: nel nostro esempio individua un miglior percorso per raggiungere z rispetto a quello che ha nella propria tabella;

	D^X	Costo via y	Costo via z
Dest.	y	2	8
	z	3	7

Tabella del nodo x

Da		Costo verso x	y	z
	x	0	2	3
	y	2	0	1
	z	7	1	0

$d_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + d_z(z)\} = = \min\{2+1, 7+0\} = 3$
➔ x invia a z tramite y a costo 3

$d_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + d_z(y)\} = = \min\{2+0, 7+1\} = 2$
➔ x invia a y tramite y a costo 2

Analogo discorso viene fatto per gli altri due nodi.

Nodo y

	D^Y	Costo via x	Costo via z
Dest.	x	2	8
	z	9	1

Tabella del nodo y

Da		Costo verso x	y	z
	x	0	2	7
	y	2	0	1
	z	7	1	0

$d_y(x) = \min\{c(y,z) + d_z(x), c(y,x) + d_x(x)\} = \min\{1+7, 2+0\} = 2$
➔ y invia a x tramite x a costo 2

$d_y(z) = \min\{c(y,z) + d_z(z), c(y,x) + d_x(z)\} = \min\{1+0, 2+7\} = 1$
➔ y invia a z tramite y a costo 1

Nodo z

	D^Z	Costo via x	Costo via y
Dest.	x	7	3
	y	9	1

Tabella del nodo z

Da		Costo verso x	y	z
	x	0	2	7
	y	2	0	1
	z	3	1	0

$d_z(x) = \min\{c(z,y) + d_y(x), c(z,x) + d_x(x)\} = \min\{1+2, 7+0\} = 3$
➔ z invia a x tramite y a costo 3

$d_z(y) = \min\{c(z,y) + d_y(y), c(z,x) + d_x(y)\} = \min\{1+0, 7+2\} = 1$
➔ z invia a y tramite y a costo 1

In ogni host sono state modificate le tabelle e quindi ogni host comunica le nuove DV a tutti i “vicini”, dove viene effettuato lo stesso procedimento di applicazione della formula di BF.

A Per prima cosa ogni router sostituisce i dati ricevuti dai DV adiacenti nel caso fossero presenti valori inferiori a quelli presenti nella propria tabella;
B applica la formula di BF per individuare il proprio percorso minimo.

Per il nodo x:
A si sostituisce un elemento nella terza riga;
B il calcolo non individua nessun percorso migliore di quelli presenti, quindi non deve inviare nessun aggiornamento.

Tabella del nodo x

		Costo verso		
		x	y	z
Da	x	0	2	3
	y	2	0	1
	z	(3)	1	0

Per il nodo y:
A si sostituisce un elemento nella prima e uno nella terza riga;
B il calcolo non individua nessun percorso migliore di quelli presenti, quindi non deve inviare nessun aggiornamento.

Tabella del nodo y

		Costo verso		
		x	y	z
Da	x	0	2	(3)
	y	2	0	1
	z	(3)	1	0

Per il nodo z:
A si sostituisce un elemento nella prima riga;
B il calcolo non individua nessun percorso migliore di quelli presenti, quindi non deve inviare nessun aggiornamento.

Tabella del nodo z

		Costo verso		
		x	y	z
Da	x	0	2	(3)
	y	2	0	1
	z	3	1	0

Una volta definite la tabelle delle distanze è immediato ricavare le tabelle di routing in quanto i pacchetti verranno indirizzati verso la porta che effettua il “primo passaggio” del cammino più breve. Nell’esempio descritto avremo:

Tabella del nodo x

		Costo verso		
		x	y	z
Da	x	0	2	3
	y	2	0	1
	z	(3)	1	0

		Costo via	
	D^X	y	z
Dest.	y	2	8
	z	3	7

Destinazione/ Network	Next Hop	Costo/ Distance
Y	y	2
Z	y	3

Tabella del nodo y

		Costo verso		
		x	y	z
Da	x	0	2	(3)
	y	2	0	1
	z	(3)	1	0

		Costo via	
	D^Y	x	z
Dest.	x	2	8
	z	9	1

Destinazione/ Network	Next Hop	Costo/ Distance
X	x	2
Z	z	1

Tabella del nodo z

		Costo verso		
		x	y	z
Da	x	0	2	3
	y	2	0	1
	z	3	1	0

		Costo via	
	D^z	x	y
Dest.	x	7	3
	y	9	1

Destinazione/ Network	Next Hop	Costo/ Distance
X	y	3
Y	y	1

Come si può osservare tutte e tre le tabelle sono uguali, quindi abbiamo avuto la convergenza e il sistema si è portato nella configurazione di minimo.
A ogni iterazione in ogni host giungono tabelle modificate con informazioni relative a host sempre più lontani: quindi al primo passo si individuano i cammini minimi tra il nodo sorgente e gli altri nodi aventi distanza unitaria, al secondo passo si trovano i cammini minimi tra il nodo sorgente e gli altri nodi con cammini di lunghezza due rami, e cosi via, si itera fino a un valore massimo di rami previsti per un cammino (generalmente questo valore è indicato con 30).

ESEMPIO 12

Come esercizio ricalcoliamo da zero le tabelle di routing nella nuova situazione di figura:

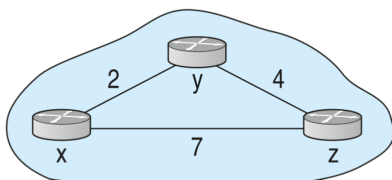


Tabella del nodo y

		Costo verso		
		x	y	z
Da	x	0	∞	∞
	y	2	0	4
	z	∞	∞	∞

Tabella del nodo z

		Costo verso		
		x	y	z
Da	x	∞	∞	∞
	y	∞	∞	∞
	z	7		0

A Entrambi spediscono gli aggiornamenti al nodo x che sostituisce i loro vettori (le loro righe) e effettua il ricalcolo per individuare i nuovi cammini minimi applicando la formula di BF:

a1: calcolo del nuovo costo minimo verso *y*

$d_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + d_z(y)\} = \min\{2+0, 7+4\} = 2$
→ x invia a y tramite y a costo 2: nessun miglioramento.

a2: calcolo del nuovo costo minimo verso z

$d_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + d_z(z)\} = \min\{2+4, 7+0\} = 6$
→ x invia a z tramite y a costo 6 che, essendo minore del valore presente nella sua tabella, comporta una nuova spedizione del suo vettore ai nodi adiacenti.

Tabella del nodo x

		Costo verso		
		x	y	z
Da	x	0	2	7
	y	2	0	4
	z	7	4	0

Tabella del nodo x

		Costo verso		
		x	y	z
Da	x	0	2	6
	y	2	0	4
	z	7	4	0

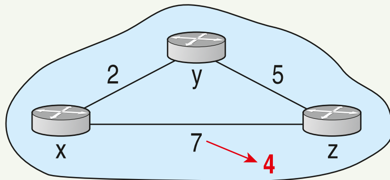
B i nodi y e z aggiornano la riga relativa al nodo x con (0,2,6) ed effettuano il ricalcolo del percorso minimo applicando la formula di BF ai nuovi valori: si può facilmente osservare che in due passaggi tutti e tre i nodi convergono alla seguente situazione finale: ▶

		Costo verso		
		x	y	z
Da	x	0	2	6
	y	2	0	4
	z	6	4	0



Prova adesso!

Supponiamo che ora venga modificato il peso dell'arco x-z come in figura:



ricalcola le tabelle applicando ai router l'algoritmo di BF.

Dato che la modifica porta un miglioramento, non è necessario partire dall'inizio ma è sufficiente effettuare il controllo sulla eventuale presenza di nuovi percorsi minimi sulle tabelle esistenti dopo aver semplicemente sostituito il vettore in arrivo con il nuovo valore.



Zoom su...

CODIFICA DELL'ALGORITMO

Di seguito è riportato un esempio di codifica in linguaggio di progetto dell'algoritmo di Bellman-Ford che utilizza le seguenti notazioni nelle istruzioni in pseudocodifica:

- c(x,y): costo del collegamento dal nodo x al nodo y; ha valore ∞ se non sono adiacenti;
- p(v): immediato predecessore di *v* lungo il cammino;
- $D^X(y,z)$: distanza da x a y, via z (che è il prossimo hop), cosi ottenuta:
- $$D^X(y,z) = c(x,z) + \min_W[D^z(y,w)];$$
  cioè dalla distanza tra x e y sommata alla minima distanza tra z e y presente nel vettore delle distanze;
- $W_D(y,w)$: vettore delle distanze tra il nodo Y e gli altri nodi;

```
per tutti i nodi adiacenti v // inizializzazione
   D^x(*,v) = ∞ // * significa "per ogni riga"
   D^x(v,v) = c(x,v)
per tutte le destinazioni y
   manda min_w D(y,w) a ogni vicino
   // inizio ciclo infinito
ripeti
   aspetta (fino a quando vedo una modifica nel costo di un
   collegamento oppure ricevo un messaggio da un vicino v)
   if (c(x,v) cambia valore)
     {cambia il costo a tutte le dest. via vicino v di d}
     per tutte le destinazioni y: D^x(y,v) = D^x(y,v) + d
   else if (ricevo mess. aggiornamento da v verso destinazione y)
     {cammino minimo da v a y è cambiato}
     {V ha mandato newval come nuovo valore per il suo min_w D^v(y,w)}
     per la singola destinazione y: D^x(y,v) = c(x,v) + newval
     if hai un nuovo min_w D^x(y,w) per una qualunque destinazione y
        manda il nuovo valore di min_w D^x(y,w) a tutti i vicini
until false
```

Problemi di instradamento

I problemi operativi del BF sono dovuti al fatto che nessuno conosce la topologia della rete e quindi le operazioni di aggiornamento vengono fatte in modo sequenziale, senza alcuna ottimizzazione e quindi ogni modifica anche di un solo peso porta spesso ad alti tempi "transitori" prima che si arrivi alla convergenza in tutta la rete.
Riportiamo sinteticamente di seguito i tre casi in cui siamo in presenza di una situazione problematica.

Black hole

Siamo in presenza di un ◀ **black hole** (buco nero) ▶ quando i pacchetti per una destinazione sono inviati a un router che li scarta, non disponendo di una route per la destinazione desiderata.

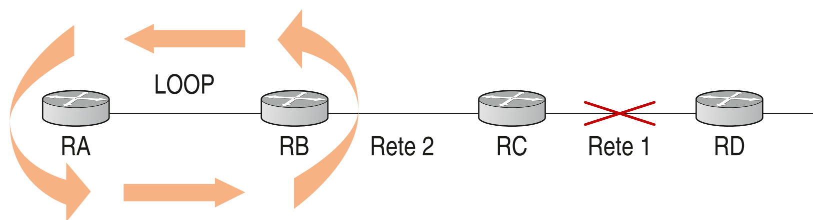


◀ **Black hole** A black hole exists on a network when a router directs network traffic to a destination that just "throws away" the traffic. ▶

Routing loop

I routing loop si possono verificare se le routing table non sono aggiornate o sono inconsistenti a causa di una lenta convergenza di tutti i router: passerà un intervallo di tempo prima che tutti si aggiornino sui nuovi percorsi in maniera corretta.

Non sempre è possibile che avvenga la convergenza il più presto possibile, come si desidererebbe: esistono anche delle situazioni in cui la rete non convergerà mai, come in caso in cui si verifichi un *routing loop*, ossia quando un pacchetto rimane vincolato a muoversi fra 2 o più router senza trovare una via d'uscita.



Vediamo un semplice esempio, supponendo che nella seguente rete si interrompa la connessione sulla Rete1.

Il Router C elimina questo percorso dalla propria routing table e al successivo update, il Router C invia la sua tabella a Router B, il quale rimuove la entry corrispondente alla Rete1.

Il Router B, osservando che Router A ha un percorso verso la Rete1 con metrica di 2, prende questo come minor percorso per raggiungere il router D e aggiorna la sua tabella mettendo come strada preferenziale verso la Rete1 il passaggio attraverso il Router A: ne aggiorna quindi la metrica di una unità portandola al valore 3.

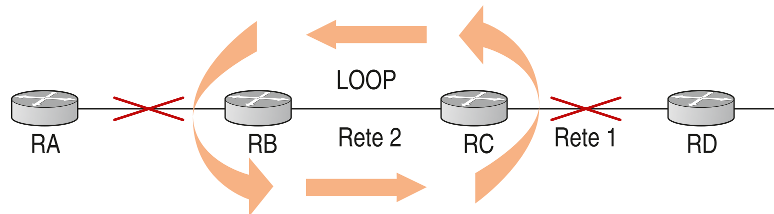
Il Router A, non appena gli perviene il vettore dal router B, "pensa" che la Rete1 sia raggiungibile attraverso Router B, e ne aggiorna la proprio tabella.

Il risultato finale è che entrambi i router inoltrano i pacchetti destinati al router D al proprio vicino, cioè A li manda a B e mi li manda ad A, continuando cioè a "rimbalzare" tra i due nodi senza arrivare mai a destinazione sino a quando viene a cessare il loro "time to live".

Questo fenomeno è chiamato effetto rimbalzo (bouncing effect) e si esaurisce solo quando la rete converge con delle coerenti tabelle di routing.

Count to infinity

Nel caso invece che avvenga la rottura di due link si potrebbe arrivare a una situazione in cui non avverrà mai la convergenza e quindi il bouncing effect diviene un loop infinito.
Si può verificare con un esempio simile al precedente che i Router B e C continuano a inviarsi aggiornamenti di tabelle incrementando di volta in volta le metriche di 2 unità fino a raggiungere l'infinito.



> Questo processo prende proprio il nome di conteggio all'infinito (counting to infinity).

Migliorie agli algoritmi di BF

Per migliorare le prestazioni dovute ai problemi esposti precedentemente vengono introdotte delle varianti agli algoritmi base di Belmann-Ford e di seguito sono riportate le più conosciute.

Numero massimo hop count

Per evitare il problema del count to infinity si introduce un valore convenzionale molto grande per l'infinito (self-correcting) e quando questo viene superato l'entry nella tabella di routing viene considerata irraggiungibile.
In altre parole si stabilisce un numero massimo di *hop count* (salti) che viene considerato come massima distanza raggiungibile nella rete, cioè il tragitto più lungo che il pacchetto può fare e una volta superato questo valore il pacchetto verrà scartato.

> È una soluzione parziale in quanto non vengono modificate le tabelle di instradamento e quindi i pacchetti nei loop si rimbalzano finché raggiungono il valore massimo occupando banda trasmissiva.

Hold down timer

Un'altra tecnica per evitare i routing loop è l'◀ **hold down timer** ▶: quando un router riceve un update da un vicino con un vettore dove viene segnalato che una rete che prima era presente ora è irraggiungibile, viene fatto iniziare un hold down timer associato a questa rete, cioè un contatore a decremento. Se il contatore raggiunge lo zero prima che un nuovo aggiornamento in arrivo dallo stesso host gli segnali che la connessione è di nuovo presente con un valore migliore a quello che ha in tabella, il router elimina la entry di quella rete in modo che risulti irraggiungibile.



> ◀ **Hold down timer** The hold down timer begins when the route is considered invalid: until the timer expires, the router will discard any subsequent route messages that indicate the route is in fact reachable. ▶

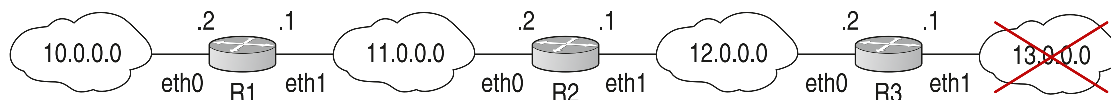
Split horizon

Si possono evitare alcuni problemi utilizzando l'algoritmo split horizon, che si fonda su una riflessione "banale":

> Se il nodo A sta instradando pacchetti per la destinazione X attraverso il nodo B, non ha senso, per B, tentare di raggiungere X attraverso A.

ESEMPIO 13

Nella rete di figura, R2 non può inviare aggiornamenti relativi alla rete 13.0.0.0 verso R3.



L'implementazione più semplice di questo algoritmo introduce una piccola variante al distance vector, dove i vettori di aggiornamento vengono differenziati a seconda del destinatario, tenendo conto del fatto che alcune spedizioni "simmetriche" sono inutili.

> Lo split horizon potrebbe però anche impedire che informazioni corrette si propaghino. Per questo alcune volte è disabilitato dall'amministratore di rete.

Poison reverse

Una forma più completa dello split horizon è quella che prende il nome di "split horizon with poison reverse" dove i nodi inviano a tutte le destinazioni il messaggio distance vector ma configurano le corrispondenti distanze a infinito per tutte le route particolari, così da considerare quelle destinazioni come irraggiungibili.

> Il metodo di split horizon, sia con ◀ **poison reverse** ▶ che senza, elimina i loop solo fra router adiacenti e non permette di eliminare tutti i tipi di loop.



◀ **Poison reverse** A technique for minimizing the time to achieve network convergence. After a connection disappears, the router advertising the connection retains the routing table entry for several update periods and specifies an infinite cost in its broadcasts. ▶

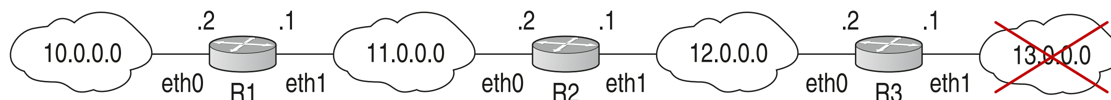
Route poisoning

Per aumentare la convergenza della rete si utilizza il route poisoning (da non confondere con l'algoritmo precedentemente descritto): questo metodo si può chiamare "precauzionale" in quanto non appena un router individua un peggioramento in una metrica ipotizza subito la presenza di un loop e mette al massimo il valore della metrica corrispondente e lo pone come irraggiungibile anche per i router adiacenti.

Invia cioè all'interno dei messaggi di aggiornamento anche le route divenute irraggiungibili con una distanza pari a 16 (unreachable).

ESEMPIO 14

Quando il router R3 si accorge che la rete 13.0.0.0 non è più raggiungibile invia al router R2 un messaggio poison con la riga 13.0.0.0 eth0 16 in modo che vengano aggiornate le tabelle come riportato in figura:



R1

Destinazione/Network	Next hop	Costo/Distance
10.0.0.0	eth0	0
11.0.0.0	eth1	0
12.0.0.0	11.0.0.2	1
13.0.0.0	11.0.0.2	2

R2

Destinazione/Network	Next hop	Costo/Distance
11.0.0.0	eth0	0
12.0.0.0	eth1	0
10.0.0.0	11.0.0.1	1
13.0.0.0	12.0.0.2	16

R3

Destinazione/Network	Next hop	Costo/Distance
12.0.0.0	eth0	0
13.0.0.0	eth0	16
11.0.0.0	12.0.0.1	1
10.0.0.0	12.0.0.1	2

Triggered updates

Le reti che implementano le tecniche di tipo distance vector inviano gli updates tra i router a intervalli di tempo regolari: per rendere più veloce la convergenza si introduce con il metodo chiamato ◀ **trigger update** ▶ la possibilità di inviare i messaggi immediatamente al momento in cui si verificano dei cambiamenti nella topologia della rete.

I trigger update vengono usati quando:
- una interfaccia cambia stato (up o down);
- una network non è più raggiungibile;
- è necessario inserire una nuova network nella tabella di routing.

Usando i triggered update contemporaneamente al route poisoning si è in grado di assicurare la diffusione nel più breve tempo possibile della presenza sulla rete di route indisponibili.

◀ **Triggered updates** Triggered updates allow a router to announce changes in metric values almost immediately rather than waiting for the next periodic announcement. ▶



Conclusioni

Sia l'algoritmo di Belmann-Ford che quello di Dijkstra convergono alla stessa soluzione e, pur essendo quest'ultimo teoricamente più complesso, in termini pratici si equivalgono.

La differenza sostanziale tra i due algoritmi, oltre al fatto che uno è statico e l'altro è dinamico, sta nel fatto che l'algoritmo di Dijkstra richiede che un nodo conosca l'intera topologia della rete e il

peso di tutti i rami mentre all'algoritmo di Bellman-Ford è sufficiente avere la conoscenza dello stato dei neighbour e quindi il colloquio si limita ai nodi adiacenti.

La quantità di traffico sulla rete è maggiore nel DV in quanto i pacchetti di dati che vengono scambiati nell'algoritmo di Dijkstra sono più piccoli dei vettori di dati; la quantità di memoria occupata per ogni nodo è pressocché equivalente nei due sistemi.

In termini di robustezza, l'algoritmo di Dijkstra offre migliori garanzie nel caso in cui si verifichi un funzionamento errato di un router in quanto in tale caso l'eventuale errore rimarrebbe circoscritto ai nodi a esso adiacenti dato che ogni nodo calcola le proprie tabelle, mentre nel Bellman-Ford un errore su di un cammino verrebbe propagato a tuti i nodi della rete in quanto il vettore delle distanze di un nodo viene utilizzato da tutti gli altri nodi fino alla convergenza dei dati.



Zoom su...

COMPLESSITÀ COMPUTAZIONALE

L'algoritmo di Bellman-Ford ha una complessità computazionale $O(n^3)$ mentre quello di Dijkstra ha una complessità $O(n^2)$ che, in applicazioni particolari, diviene $O(n \log n)$: quindi l'algoritmo di Dijkstra è più conveniente.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 La classificazione degli algoritmi di routing in base alla topologia prevede:

a) algoritmi distribuiti, isolati, centralizzati
b) algoritmi locali, remoti, isolati
c) algoritmi distribuiti, isolati, locali
d) algoritmi remoti, isolati, centralizzati

2 Un vettore distanza è composto da:

a) destinazione, mittente
b) destinazione, costo
c) costo, mittente
d) mittente, destinazione, costo

3 Quale tra le seguenti è chiamata Formula di Bellman-Ford?

a) $d_x(y) = \min_v \{c(x,v) + d_v(y)\}$
b) $d_y(x) = \min_v \{c(x,v) + d_v(y)\}$
c) $d_x(y) = \min_v \{c(x,y) + d_v(x)\}$
d) $d_y(x) = \min_v \{c(y,v) + d_v(x)\}$

4 La formula di Bellman-Ford (indicare le voci errate):

a) viene eseguita su ciascun host
b) è ricorsiva
c) permette di aggiornare i distance vector
d) permette di aggiornare le tabelle in parallelo

5 Per migliorare l'algoritmo di BF si possono utilizzare le seguenti varianti:

a) si stabilisce un numero massimo di hop count
b) si introduce un hold down timer associato a una rete irraggiungibile
c) si utilizza l'algoritmo split Horizon
d) si introduce il poison reverse in caso di bouncing effect
e) si introduce il trigger update

6 I trigger update vengono usati quando (indicare la voce errata):

a) una interfaccia cambia stato (up o down)
b) una network non è più raggiungibile
c) si è in presenza di un hold down timer
d) è necessario inserire una nuova network nella tabella di routing

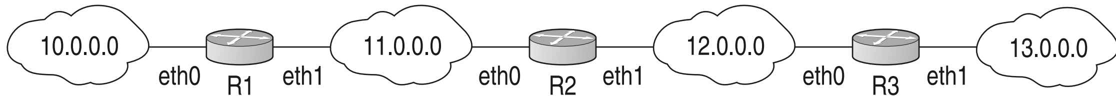
Test vero/falso

1 L'algoritmo di Bellman-Ford è di tipo asincrono. V F
2 L'algoritmo di Bellman-Ford è di tipo non adattativo. V F
3 Con neighbour si intendono gli host confinanti, direttamente raggiungili con un unico hop. V F
4 Il triggered update è l'aggiornamento che viene inviato ogni volta che si modifica la tabella. V F
5 Siamo in presenza di un "buco nero" se un router scarta i pacchetti per una destinazione. V F
6 I routing loop si possono verificare se le routing table non sono aggiornate. V F
7 I routing loop si possono verificare se le routing table sono inconsistenti. V F
8 Il nome di conteggio all'infinito è dovuto alla rottura di uno o più link. V F
9 Solo lo split horizon con poison reverse elimina i loop solo fra router adiacenti. V F
10 L'algoritmo di Bellman-Ford ha una complessità computazionale $O(N^2)$. V F

Verifichiamo le competenze

Esercizi

1 Data la rete di figura



determinare le tabelle di routing mediante l'algoritmo di Bellman-Ford.

Istante t0

R1

Network	Next hop	Distance

R2

Network	Next hop	Distance

R3

Network	Next hop	Distance

Istante t1

R1

Network	Next hop	Distance

R2

Network	Next hop	Distance

R3

Network	Next hop	Distance

Istante t2

R1

Network	Next hop	Distance

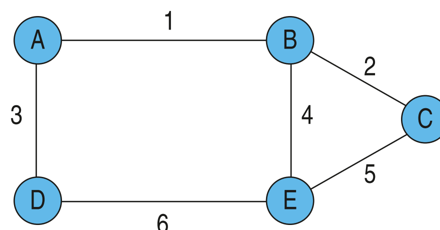
R2

Network	Next hop	Distance

R3

Network	Next hop	Distance

2 Data la seguente rete dove si considerano i pesi unitari e sugli archi sono indicati i numeri dei link, calcolare le tabelle di routing di ogni nodo applicando l'algoritmo di Bellman-Ford.



Da A a:	Link	Costo
A		
B		
D		
C		
E		

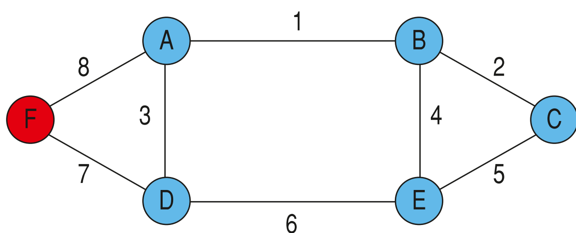
Da B a:	Link	Costo
B		
A		
D		
C		
E		

Da C a:	Link	Costo
C		
B		
A		
E		
D		

Da D a:	Link	Costo
D		
A		
B		
E		
C		

Da E a:	Link	Costo
A		
B		
C		
D		
E		

3 Alla rete dell'esercizio precedente viene aggiunto un host: effettuare l'aggiornamento delle tabelle di routing utilizzando l'algoritmo di BF.



La costruzione della tabella del nodo F avviene in base ai vettori dei suoi neighbour e, a parità di metriche, viene utilizzato l'entry di quella che "arriva per primo", cioè a parità di valori non viene sostituito quello che è già presente nella tabella.

Da A a:	Link	Costo
A		
B		
D		
C		
E		
F		

Da B a:	Link	Costo
B		
A		
D		
C		
E		
F		

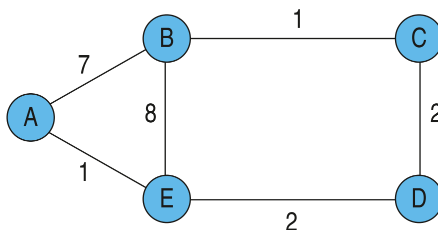
Da C a:	Link	Costo
C		
B		
A		
E		
D		
F		

Da D a:	Link	Costo
A		
B		
D		
C		
E		
F		

Da E a:	Link	Costo
B		
A		
D		
C		
E		
F		

Da F a:	Link	Costo
F		
A		
B		
C		
D		
E		

4 Data la rete di figura



determinare la tabella delle distanze e la routing table per il nodo E.

		Costo via		
	D^E	A	B	D
Destinazione	A			
	B			
	C			
	D			

Destinazione	Next hop	Costo
A		
B		
C		
D		

5 Una rete possiede cinque router (A, B, C, D ed E) che stabiliscono le proprie tabelle di instradamento in base a un algoritmo di tipo Distance Vector. Il nodo C possiede la seguente tabella di instradamento:

Da C a:	Link	Costo
A	B	4
B	B	2
D	D	7

Specificare le modifiche alla tabella di instradamento dopo la ricezione di ciascuno dei tre messaggi seguenti, ricevuti nell'ordine indicato:

Da B a:	Costo
A	1
C	2
D	10
E	20

Da B a:	Costo
A	8
B	12
C	7
E	5

Da B a:	Costo
A	3
B	2
D	10
E	5

Dopo il primo DV

Da C a:	Link	Costo
A		
B		
D		
E		

Dopo il secondo DV

Da C a:	Link	Costo
A		
B		
D		
E		

Dopo il terzo DV

Da C a:	Link	Costo
A		
B		
D		
E		

Lezione 6

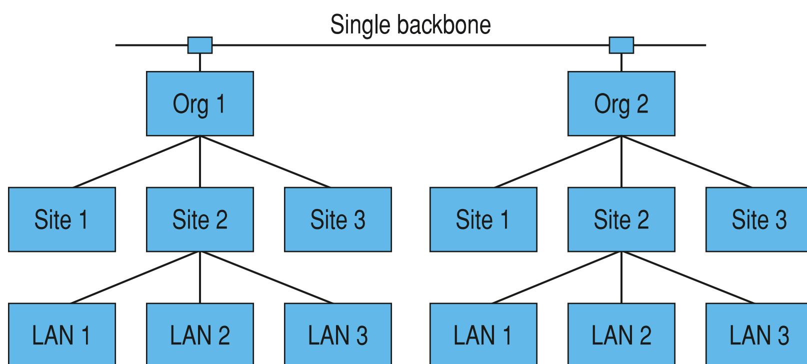
ROUTING GERARCHICO

IN QUESTA LEZIONE IMPAREREMO...

- il concetto di autonomous system (AS)
- i protocolli IGP: RIP e OSPF
- un protocollo EGP: il BGP

Introduzione

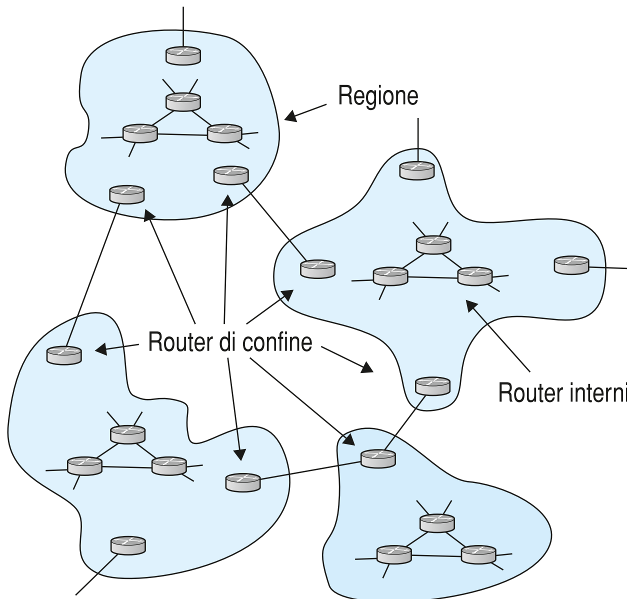
Negli anni '80 l'architettura di Internet era molto semplice ed era costituita da un'unica backbone e ogni rete fisica era collegata alla backbone da un router che conosceva le rotte per tutte le reti fisiche.

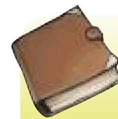


Ben presto la rete si è trasformata e si è strutturata in un insieme di diverse dorsali connesse tra loro (peer backbone); inoltre, con l'aumento del numero di nodi, si è arrivati a una situazione in cui qualunque strategia di routing globale diventa inefficace.

Si ricorre perciò al routing gerarchico dove l'internetwork viene suddivisa in varie regioni all'interno delle quali si applica una strategia di routing, anche diversa tra le regioni stesse: queste regioni sono anche chiamate domini di routing o autonomous system (AS) e sono in pratica dei raggruppamenti logici di computer.

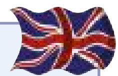
All'interno degli ◀ **autonomous system** ▶ i router possono interagire tra loro, ma solo tra quelli che appartengono allo stesso dominio: prendono il nome di router interni; per poter comunicare con un altro dominio è necessario inviare i dati a particolari router, detti router di confine, che sono "specializzati" nel comunicare con le altre regioni.



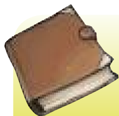


AUTONOMOUS SYSTEM

Si definisce "sistema autonomo" (AS) una porzione di rete (insieme di host, router e sottoreti) amministrata da un unico gestore.



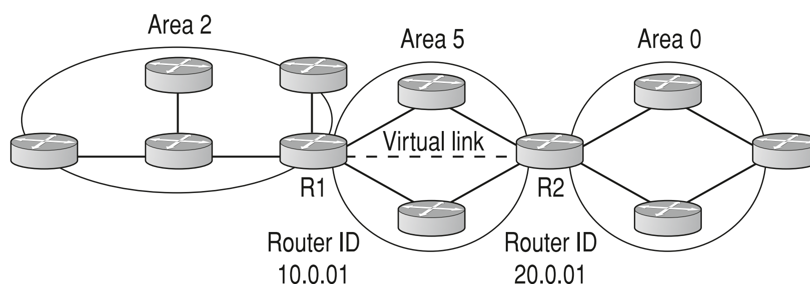
◀ **Autonomous system** An autonomous system (AS) is the unit of router policy, either a single network or a group of networks that is controlled by a common network administrator (or group of administrators) on behalf of a single administrative entity (such as a university, a business enterprise, or a business division). ▶



BACKBONE AREA

Con backbone area (o area 0) si intende una area che racchiude tutti i router di confine e tutte le reti che non sono di nessuna area specifica: serve per mettere in comunicazione tutte le aree tra di loro.

Tutte le aree devono essere fisicamente connesse alla backbone area che, però, può anche non essere contigua oppure non direttamente limitrofa a tutte le aree (come riportato nell'esempio della figura seguente): in questo caso la connettività viene realizzata con un ◀ **Virtual Link** ▶.

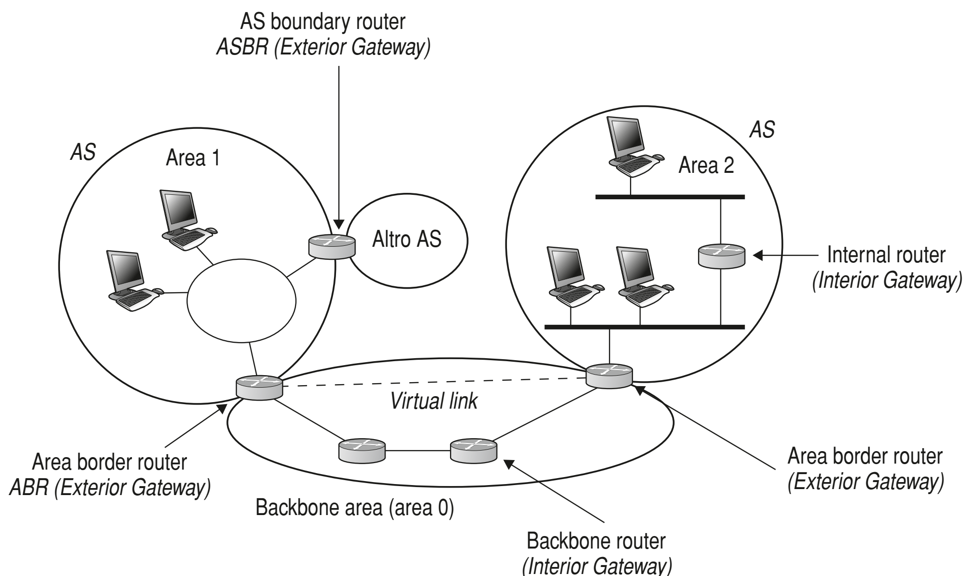




◀ **Virtual link** In some cases where the physical connection is not possible, you can use a virtual link to connect to the backbone through a non-backbone area. You can also use virtual links to connect two parts of a partitioned backbone through a non-backbone area. ▶

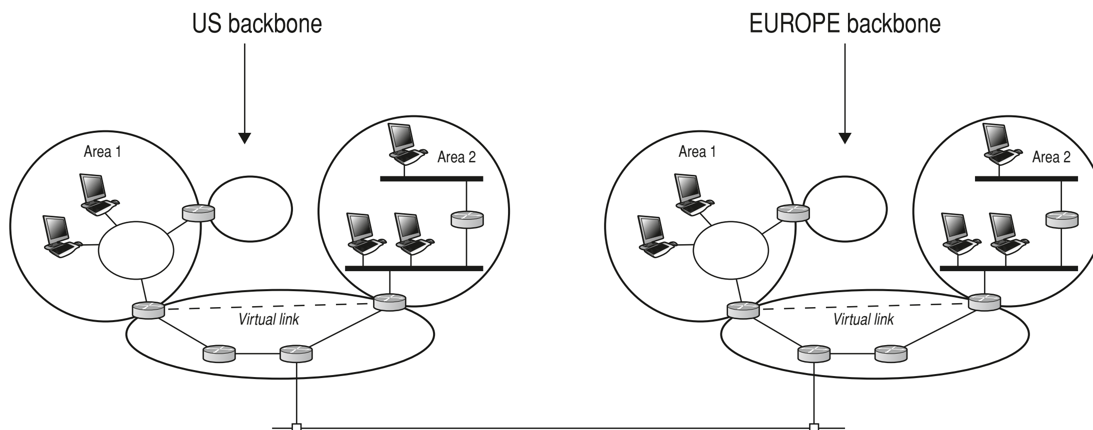
Possiamo quindi effettuare un'ulteriore classificazione dei router di confine, a seconda che siano o meno in contatto con la backbone area:

- router di confine dell'area, o Area Border Router (ABR): sono di confine tra più aree e contengono una copia per ogni area più quella per l'area di backbone;
- router di confine dell'AS, o Boundary Router (ASBR): router di confine tra AS.



Si crea quindi una gerarchia di router: quella sopra descritta è su due livelli, ma generalmente due soli livelli non sono sufficienti.

Internet, per esempio, è una collezione di AS interconnessi mediante alcune backbone principali costituiti da linee ad alta velocità che connettono un insieme di reti regionali (USA) e nazionali (Europa e resto del mondo) ciascuna composta da reti locali.



Tassonomia dell'internetworking

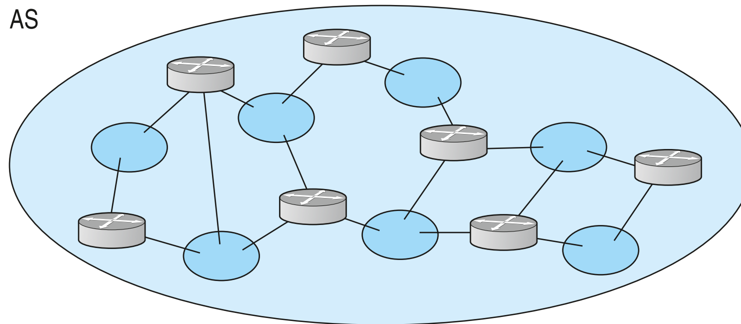
Introduciamo una tassonomia delle reti in modo da formalizzare l'utilizzo della terminologia specifica dell'internetworking.



AUTONOMOUS SYSTEM AS

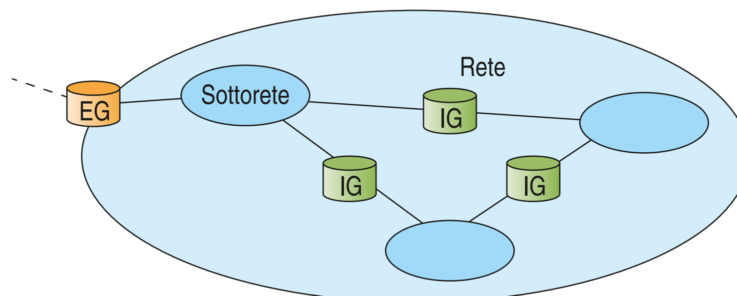
La parte di rete IP gestita da una organizzazione (un unico gestore) è chiamata Sistema Autonomo.

Ogni AS è univocamente identificato tramite un nome composto da 16 bit e all'interno della sua organizzazione definisce in maniera coerente (e autonoma) le politiche di instradamento.



ESEMPIO 15

Per esempio, un AS è la rete di un ISP (Internet Service Provider): i router all'interno di esso sono detti Interior Gateway (IG) o semplicemente IR (Interior Router) mentre quelli di collegamento con altri AS sono detti Exterior Gateway (EG) o semplicemente ER (Exterior Router).



Gli IR (Interior Router o intraAS) si scambiano informazioni di instradamento tramite uno dei seguenti protocolli, detti Interior Gateway Protocol (IGP):

- protocolli che utilizzano il Distance Vector:
  - RIP: Routing Information Protocol;
  - IGRP: Interior Gateway Routing Protocol (proprietario di Cisco Systems);
  - EIGRP: Enhanced IGRP;
- protocolli che utilizzano il Link State:
  - OSPF: Open Shortest Path First;
  - Integrated IS-IS.

Gli ER (Exterior Router o interAS) si scambiano informazioni di instradamento tramite altri protocolli, detti Exterior ◀ Gateway ▶ Protocol (EGP), tra i quali ricordiamo:
- EGP: Exterior Gateway Protocol (oggi obsoleto);
- BGP: Border Gateway Protocol;
- IDRP: Inter-Domain Routing Protocol.



◀ **Gateway** A node on a network that serves as an entrance to another network: the gateway is the computer that routes the traffic from a workstation to the outside network that is serving the Web pages. ▶



Zoom su...

TIPI DI AS

Possiamo introdurre una classificazione anche per gli AS:
- se è presente solo un router di confine prende il nome di stub o single-homed, ed è il caso tipico di piccole organizzazioni;
- nel caso in cui siano presenti più router di confine prende il nome di multi-homed e può essere di due tipi:
  – transit (provider): accetta di essere attraversato da traffico diretto ad altri AS;
  – non-transit (grandi corporate): non accetta di essere attraversato da traffico diretto ad altri AS.

Interior Gateway Protocol (IGP)

Tra i protocolli utilizzati dai router interni all'AS noi descriveremo solamente i primi due riportati nell'elenco precedente, cioè il RIP (sia nella versione 1 che nella versione 2) e l'OSPF.

RIP v1

Viene descritto nella RFC 1058 e si basa su un algoritmo della famiglia distance vector (algoritmo di Bellmann-Ford) dove viene utilizzata una metrica elementare che conta semplicemente il numero di hop con valore massimo 15: il numero 16 sta a indicare l'infinito.

È molto semplice da implementare e realizzare e tra i protocolli di routing dinamico è forse il più diffuso soprattutto per reti di piccole dimensioni, nonostante la lentezza della convergenza.

Il formato del messaggio RIP è riportato nello schema seguente, dove a sinistra è indicato il numero dei byte che compongono ogni singolo dato:

bytes			
	1	command	Request (1)/Response (2)
	1	version	1/2
	2	0	reserved
	2	address family id. (IP-2)	Ripetuto fino a 25 volte
	2	0	
	4	address	
	4	0	
	4	0	
	4	metric	
		address family id.	
		address	
		...	

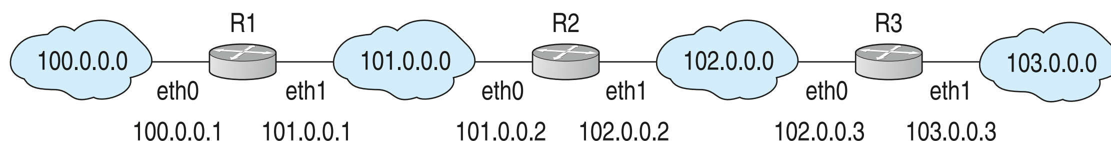
Il blocco di dati viene ripetuto 25 volte in quanto questo è il numero di sottoreti raggiungibili tramite il router che lo ha inviato. Come vedremo in seguito, i messaggi RIP vengono incapsulati nel protocollo UDP: viene utilizzata per questo servizio la porta 520.
Una caratteristica di questo protocollo è che viene regolato da un ◀ route timer ▶: i router che condividono dei link si scambiano aggiornamenti (UPDATE) sulle loro tabelle di instradamento (il vettore delle distanze DV sopra descritto) ogni 30 secondi circa: questo messaggio prende il nome di RIP advertisement.
Se un router non riceve notizie da uno dei suoi vicini per più di 180 secondi considera il link come non più utilizzabile, quindi la route non è più valida e la sua metrica viene posta al valore di infinito (16) nella tabella di forwarding locale: questa notizia deve essere diffusa, e quindi viene propagata ai router vicini che la diffonderanno a loro volta su tutta la rete. Se inoltre non arrivano messaggi di advertisement da una route per 240 secondi questa route viene cancellata dai DV e viene eventualmente inserita successivamente dalla analisi dei dati dei DV arrivati in seguito dagli altri ruoter.



◀ Route timer I valori dei timer sono settati di default, ma possono essere cambiati; hanno i seguenti nomi e valori:
- route update timer = 30 s;
- route invalid timer = 180 s;
- route flush timer = 240 s. ▶

ESEMPIO 16

Vediamo un semplice esempio di funzionamento del protocollo RIP all'accensione della rete di figura:



Per ogni router sono indicati gli indirizzi IP delle relative interfacce: all'accensione dei router nella tabelle di routing di ciascun host sono presenti solo le entry relative ai vicini:

R1

Network	Next hop	Distance
100.0.0.0	eth0	0
101.0.0.0	eth1	0

R2

Network	Next hop	Distance
101.0.0.0	eth0	0
102.0.0.0	eth1	0

R3

Network	Next hop	Distance
102.0.0.0	eth0	0
103.0.0.0	eth1	0

Ciascuno di essi invia un messaggio di UPDATE verso i propri vicini all'istante di tempo t_0.

Il router R1 riceve dal router R2 i due dati relativi alle entry presenti nella sua tabella ed effettua la fusione con la propria tabella di routing, aggiungendo la sola entry mancante, la strada verso 102.0.0.0, dato che la rete 101.0.0.0 è già da lui raggiungibile con la stessa metrica: la tabella diviene la seguente:

R1

Network	Next hop	Distance
100.0.0.0	eth0	0
101.0.0.0	eth1	0
102.0.0.0	101.0.0.2	1

La distanza per raggiungere questa nuova rete è settata al valore 1, dato che per raggiungerla bisogna prima inviare il pacchetto al router R2.

Analogo discorso viene fatto per il router R3 che riceve i dati da R2, modificando la propria tabella nella seguente:

R3

Network	Next-hop	Distance
102.0.0.0	eth0	0
103.0.0.0	eth1	0
101.0.0.0	102.0.0.2	1

Il router R2 riceve le due tabelle da R1 ed R3 aggiungendo due entry:

R2

Network	Next-hop	Distance
101.0.0.0	eth0	0
102.0.0.0	eth1	0
100.0.0.0	101.0.0.1	1
103.0.0.0	102.0.0.3	1

Ciascuno invia successivamente un messaggio di UPDATE verso i propri vicini all'istante di tempo t_1 con i nuovi valori appena aggiornati: alla sua ricezione i router R1 ed R3 aggiungono una nuova riga alla propria tabella, naturalmente dopo aver incrementato di un hop la distanza:

R1

Network	Next hop	Distance
100.0.0.0	eth0	0
101.0.0.0	eth1	0
102.0.0.0	101.0.0.2	1
103.0.0.0	101.0.0.2	2

R2

Network	Next hop	Distance
101.0.0.0	eth0	0
102.0.0.0	eth1	0
100.0.0.0	101.0.0.1	1
103.0.0.0	102.0.0.3	1

R3

Network	Next hop	Distance
102.0.0.0	eth0	0
103.0.0.0	eth1	0
101.0.0.0	102.0.0.2	1
100.0.0.0	102.0.0.2	2

Il successivo UPDATE non porta più nessuna modifica.



Zoom su...

IMPLEMENTAZIONE DI RIP

Il protocollo RIP può essere implementato su un'apparecchiatura dedicata oppure con un processo utente di bassa priorità (un demone, per utilizzare la terminologia UNIX, di livello 7), che colloquia con gli altri router tramite un socket UDP (porta 520) per aggiornare le strutture interne del kernel.

È anche possibile implementare RIP sui calcolatori dove sono già presenti altri servizi, come il firewall.

Il RIP utilizza tre vettori, che riportiamo in figura con la seguente notazione:

- M: numero di reti a cui il nodo x è direttamente connesso;
- w(x,i): costo associato al ramo uscente dal nodo x verso la rete i (1 ≤ i ≤ M);

- N: numero di reti raggiungibili;
- L(x,j): stima della lunghezza del cammino minimo dal nodo x alla rete j;
- R(x,j): next router nel cammino a minima lunghezza dal nodo x alla rete j.

$$W_x = \begin{vmatrix} w(x,1) \\ \vdots \\ w(x,M) \end{vmatrix} \qquad L_x = \begin{vmatrix} L(x,1) \\ \vdots \\ L(x,N) \end{vmatrix} \qquad R_x = \begin{vmatrix} R(x,1) \\ \vdots \\ R(x,N) \end{vmatrix}$$

Vettore dei costi dei rami uscenti dal nodo — Vettore delle distanze minime verso le altre reti — Vettore next hop verso le altre reti

A ogni scadenza del *route update* timer avviene lo scambio del vettore L_x tra nodi adiacenti e in base a esso ogni nodo *x* aggiorna i suoi vettori nel seguente modo:

$$L(x, j) = \min_{y \in A} w(x,N_{xy})+L(y,j)$$

$$R(x, j) = y$$

dove
- A: insieme dei nodi vicini al nodo x;
- N_{xy}: rete che interconnette i nodi x e y.

RIPv2

La versione 2, descritta nella RFC 1723, modifica il record che viene trasferito nel DV aggiungendo anche la netmask, in modo da permettere una “autenticazione” dell’host.

4	address
4	netmask
4	metric

OSPF: Open Shortest Path First

È un protocollo usato per l’instradamento intra-AS che supera le limitazioni del protocollo RIP consentendo una maggiore velocità di convergenza in quanto adotta un protocollo di tipo Link State dove ogni router calcola lo spanning tree a partire dal grafo rappresentativo della rete mediante il metodo di Dijkstra per determinare il percorso di minor costo.

Open indica che le specifiche del protocollo di instradamento sono disponibili al pubblico (a differenza di altri protocolli come EIGRP della Cisco).

OSPF supporta routing gerarchico ed è descritto dall’IETF con la RFC 1247 (1991), la RFC 1583 (1994) e la versione più recente, la 2, è definita nella RFC 2328.

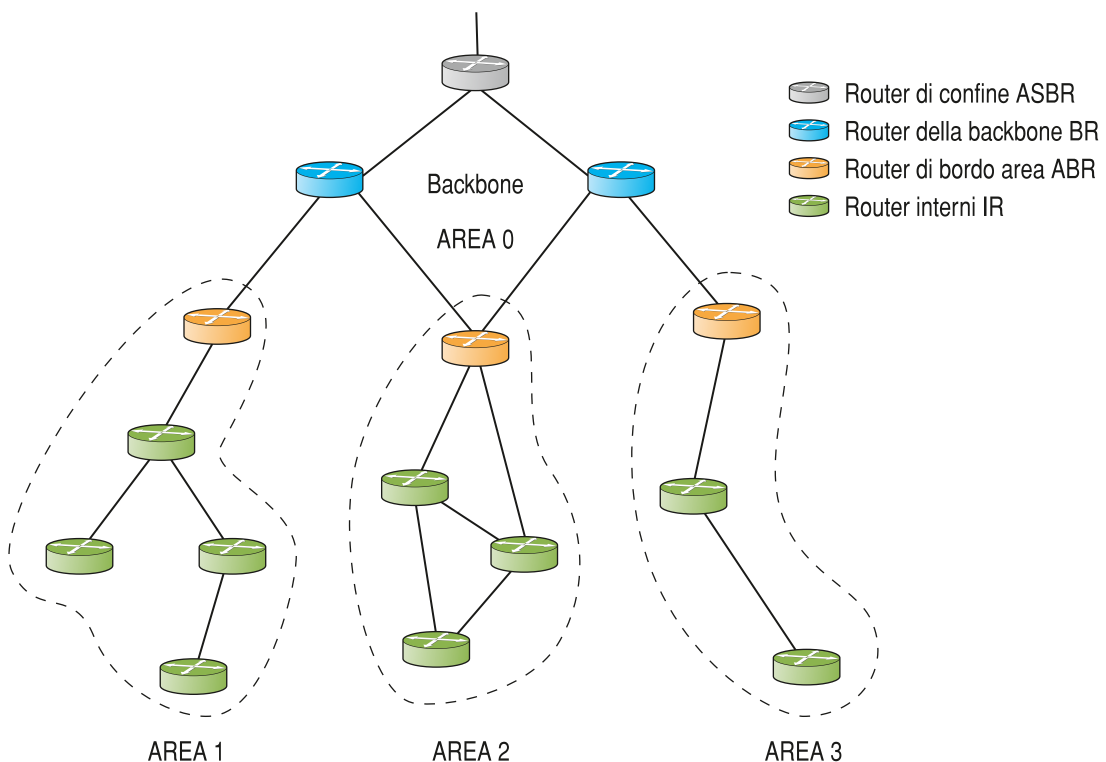
L’OSPF definisce una gerarchia su due livelli, “area locale” e “dorsale”, che costituiscono i domini di instradamento: i messaggi di link-state vengono propagati solo all’interno dell’area locale di ciascun nodo.
In ogni area è quindi presente un proprio database topologico uguale per ogni router interno all’area stessa, dove ogni nodo conosce solo la direzione (shortest path) verso le reti presenti nelle altre aree e inoltra i DV verso i router di confine, che scambiano informazioni con i router di altri sistemi autonomi, diffondendo in ciascuna area un riassunto delle informazioni raccolte nell’altra area.

Ogni router OSPF descrive la propria topologia (interfacce attive) tramite dei pacchetti denominati Link State Advertisement (LSA): li invia in broadcast ogni volta che c'è un cambiamento nello stato di un link ma anche periodicamente (almeno una ogni 60 minuti), anche se lo stato del link non è cambiato, per confermare che il link è attivo (in questo caso viene inviato un messaggio di HELLO, come vedremo in seguito).

I due livelli di routing prendono anche il nome di:
- routing intra-area per i router interni;
- routing inter-area per i router di confine.

Questa struttura gerarchica utilizzata da OSPF si adatta bene a reti di grandi dimensioni in quanto riduce il traffico di routing dato che ogni area si comporta come una rete autonoma e quindi non inoltra messaggi al di fuori di essa se non quelli destinati alla altre aree.

Anche i router sono stati classificati in base alla loro posizione nella rete, come indicato dalla seguente figura:



- router interni (IR: Internal router): si trovano in aree non dorsali ed effettuano soltanto instradamento interno al sistema autonomo.
- router di confine d'area (ABR: Area Border Router): appartengono sia a un'area generica sia alla dorsale (backbone).
- router di dorsale (BR: Backbone Router): effettuano l'instradamento all'interno della dorsale, ma non sono router di confine.
- router di confine (ASBR: Autonomous System Boundary Router): scambiano informazioni con i router di altri sistemi autonomi e possono, per esempio, utilizzare BGP per l'instradamento tra AS.

Un sistema autonomo OSPF può essere configurato in aree che eseguono diversi algoritmi OSPF; ciascun router in un'area invia i pacchetti agli host di destinazione presenti in un'altra area mediante uno o più router di confine, che si fanno carico dell'instradamento dei pacchetti indirizzati all'esterno. Ci sarà poi una particolare area, chiamata backbone area, configurata per rappresentare una dorsale di collegamento, il cui compito sarà quello di instradare il traffico tra le altre aree del sistema autonomo.
La dorsale contiene sempre tutti i router di confine dell'AS ma non necessariamente soltanto quelli in quanto potrebbe contenere dei router interni BR senza "contatti" con nessun AS.

Vengono inoltre definiti due router con incarichi particolari: il Designated Router (DR) e il Backup Designated Router (BDR).

- Il Designer Router ha la funzione di "concentratore di informazioni" che mantiene i dati topologici della rete: se avviene un cambiamento in un database di un qualunque router, le modifiche non vengono inviate a tutti i router ma solo a DR che provvede poi ad aggiornare gli altri router, evitando l'overhead che si genererebbe dal traffico di pacchetti OSPF tra tutte le coppie di router: quindi quando un router R deve propagare un LSA sulla LAN, lo invia al DR mediante l'indirizzo multicast data-link: DR raccoglie inoltre le conferme esplicite della ricezione degli LSA da parte degli altri router.
- Il Backup Designated Router ha la funzione indicata dal suo nome, cioè quello di tenere una copia in mirroring del DR.

L'elezione del DR e del BDR avviene in ogni rete multiaccesso, per cui un router potrebbe essere un DR su una interfaccia ma non su un'altra.

Sottoprotocolli di OSPF

OSPF è costituito da 3 "sotto-protocolli":

- protocollo Hello: verifica che i link siano attivi mediante la trasmissione periodica (ogni *hello-interval* secondi ai router vicini) di pacchetti particolari ed elegge il Designated Router e Backup Router;
- protocollo Exchange: alla prima connessione di un router con un altro inizializza il proprio Link State Database (database topologico) comunicando con messaggi di tipo database description;
- protocollo Flooding: permette di gestire le variazioni incrementali e successive comunicando i dati sulle metriche agli altri router mediante messaggi di tipo link status.

La struttura della intestazione di un messaggio OSPF, che è composto da 24 byte, è la seguente:

<table>
<tr><td>1 4</td><td>8</td><td>16 19 32</td></tr>
<tr><td>Version (2)</td><td>Type</td><td>Message Lenght</td></tr>
<tr><td colspan="3">Router ID</td></tr>
<tr><td colspan="3">Area ID</td></tr>
<tr><td colspan="2">Checksum</td><td>Authentication Type</td></tr>
<tr><td colspan="3">Authentication</td></tr>
<tr><td colspan="3">Authentication</td></tr>
</table>

Dove:

A il campo *type* specifica il tipo di messaggio OSPF che può essere:

- 1 = Hello;
- 2 = database description (DD);
- 3 = link status request;

- 4 = link status update;
- 5 = link status acknowledgement;

B Message Lenght indica la lunghezza del pacchetto (in byte);
C il campo Router ID contiene l'indirizzo IP scelto per identificare il router mittente;
D il campo Area ID codifica l'area di appartenenza, dove il valore 0 è riservato per la backbone area mentre per le altre aree si utilizza un IP network number;
E checksum che viene calcolato sull'intero pacchetto escluso il campo authentication;
F il campo Authentication type può essere 0 (No authentication), 1 (Simple authentication) o 2 (Cryptographic authentication);
G authentication è un campo di 8 byte contenente informazioni utili in caso di autenticazione.

Alla intestazione segue un insieme di informazioni specifiche per ogni tipo di messaggio.

> I pacchetti di database description, link state update e link state acknowledgment possono contenere i pacchetti LSA (Link State Advertisement), che sono di 5 tipi (descritti in seguito), con le informazioni topologiche della rete e vengono trasmessi secondo le regole del selective flooding.

Protocollo Hello

Il protocollo "Hello" viene utilizzato per 2 scopi ben precisi:

- eleggere il Designed Router e il Backup Designed Router;
- verificare l'operatività dei link.

> I pacchetti "Hello" vengono trasmessi solo tra nodi vicini e mai propagati.

Viene definita quindi una classe, come estensione di OSPF Packet, OSPF Hello packet, con la seguente struttura:

0	8	16	31
Network Mask			
Hello Interval		Options	Priority
Dead Interval			
Designated Router			
Backup Designated Router			
Neighbor			
...			
Neighbor			

- NetworkMask: netmask associata all'interfaccia da cui viene emesso il pacchetto.
- HelloInterval: indica ogni quanti secondi il router invia messaggi di Hello.
- Options: vengono definiti solo gli ultimi 2 bit, che indicano se il router è in grado di inviare e ricevere route esterne e se l'interfaccia appartiene a una stub area e di gestire il routing TOS.
- RouterPriority: serve per l'elezione del Designed Router e viene settato dall'amministratore; ciascun router è configurato con una priorità, che può variare tra 0 e 255: il router che ha priorità più alta viene eletto DR.
- DeadInterval: contiene l'intervallo di tempo di validità dei pacchetti di Hello ricevuti, oltrepassato il quale gli Hello ricevuti vengono ritenuti decaduti.
- DesignatedRouter: contiene l'indirizzo IP del DR.

- BackupDesignatedRouter: contiene l'indirizzo IP del router che diventa il principale se il router principale in uso ha un problema o si guasta.
- Neighbor: lista di router_ID da cui è stato ricevuto il pacchetto di Hello negli ultimi DeadInterval secondi.

Protocollo exchange

La sincronizzazione iniziale dei database di due router OSPF che per la prima volta stabiliscono la connessione su un link punto a punto avviene tramite il protocollo exchange mediante la comunicazione con il DR (o il DBR): successivamente gli aggiornamenti vengono comunicati con il protocollo flooding che si occupa di mantenere sincronizzati i database.

Il protocollo exchange utilizza un record, il Database Description Packet, riconosciuto dal valore 2 presente nel campo Type dell'header.

Il tracciato record è il seguente:

0	16	24	29	30	31
0	Options	0	I	M	MS
DD Sequence Number					
Link State Type					
Link State ID					
Advertising Router					
Link State Sequence Number					
LS Checksum	LS Age				

...

dopo il campo Options, uguale al protocollo Hello, sono presenti tre flag:
- I: Initialize
- M: More
- MS: Master - Slave (1= Master)

> Prima di iniziare a comunicare col protocollo Exchange tra i due router è necessario selezionarne uno come "master" e uno "slave": solo successivamente i due routers si scambieranno la descrizione dei propri database (protocollo asimmetrico).

DD Sequence Number che è il numero di sequenza del pacchetto Database Description mentre i campi successivi sono la descrizione dell'header di un LSA.

Vediamo come vengono utilizzati i flag e il campo DD descrivendo un utilizzo del protocollo:

A il router che vuole iniziare la procedura di exchange emette un pacchetto vuoto Database Description con il numero di sequenza settato a un valore arbitrario;

B il router che risponde emette anch'esso un pacchetto DD di Acknowledgment, con lo stesso numero di sequenza e i bit I ed M settati a 1 e MS a 0 (slave);

C ora il primo router può iniziare a inviare le descrizioni da lui possedute e quindi emette pacchetti DD con I settato a 0, M ed MS settati a 1: alla trasmissione dell'ultimo pacchetto setterà M con valore 0.

D a ogni ricezione lo slave risponde con un pacchetto DD di Ack con M settato a 1: la procedura termina quando lo slave riceve un pacchetto finale (M=0) e risponde anch'esso con pacchetto con M=0.

Protocollo flooding

Il protocollo flooding viene utilizzato per diffondere a tutta la rete il nuovo stato di un link attraverso il pacchetto di Link State Update, nelle due situazioni seguenti:
- quando avviene un cambiamento di stato del link;
- allo scadere di un timer (normalmente 60 min).

> Tutti gli LSA richiesti dai vari router vengono inviati attraverso il protocollo di flooding.

Il pacchetto Link State Update viene individuato dal valore 4 del campo Type dell'header comune, ed è formato da 2 campi:

0 … 31
Number of Advertisements
LSA

- Number of Advertisement: è il numero di LSA che sono presenti nel pacchetto in esame in quanto è possibile inviare più LSA contemporaneamente;
- LSA: è il Link State vero e proprio.

L'avvenuta ricezione di un pacchetto Link State Update avviene mediante la trasmissione di un pacchetto di Link State Acknowledgment con valore 5 nel campo Type dell'header comune.

Pacchetto LSA: Link State Advertisement

Il pacchetto LSA che ogni router OSPF utilizza per descrivere la propria topologia è così costituito:

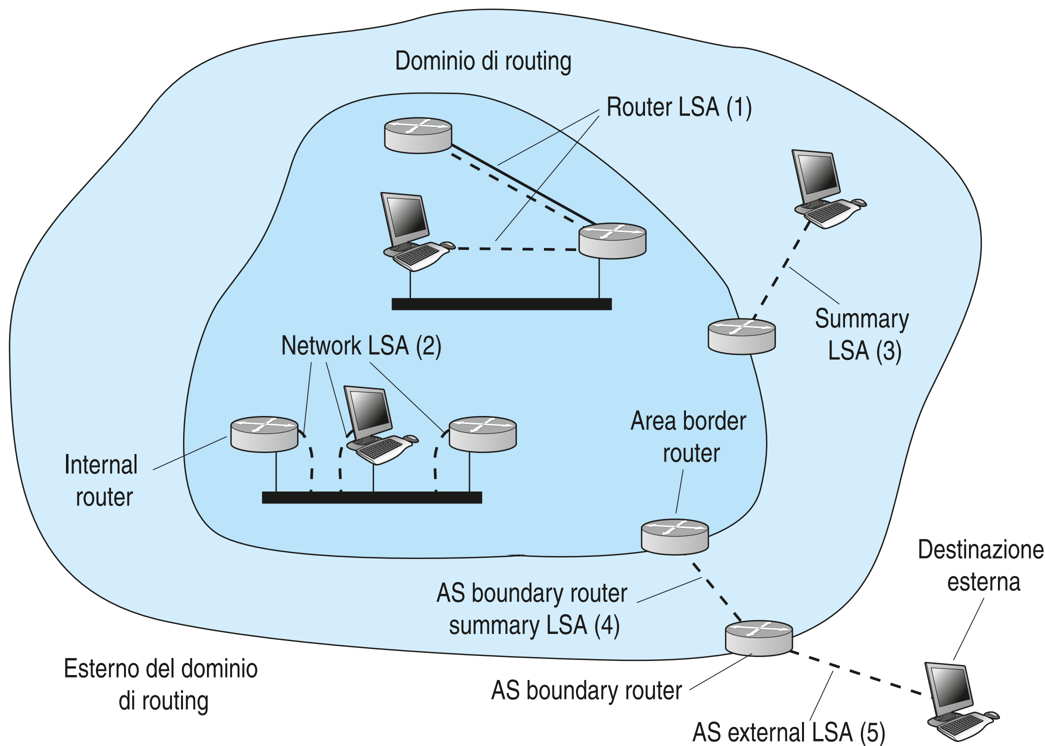
Link State age	Reserved	E	T	Link State type
Link State ID				
Advertising router				
Link State sequence number				
Link State Checksum	Lenght			

Gli LSA vengono propagati in tutta la rete tramite la tecnica del flooding: ogni router che riceve un LSA lo invia su tutte le interfacce a eccezione di quella da cui l'ha ricevuto.

Descriviamo i campi più importanti:

1. LSA age: è il tempo espresso in secondi da quando il LSA è stato generato e un router deve ripetere gli LSA da esso originariamente generati ogni 1800 s; un LSA deve essere cancellato dopo 3600 s.
2. LSA type: indica il tipo del pacchetto, che può essere uno dei cinque possibili:
   1) Router links
   2) Network links
   3) Summary link (IP network)
   4) Summary link (ASBR)
   5) AS external link
3. Link State ID: insieme all'Advertising Router permette di identificare univocamente il LSA.
4. Advertising Router: contiene l'identificativo del router che ha prodotto il LSA.
5. LSA sequence number: viene usato per distinguere versioni successive dello stesso LSA.

Descriviamo dettagliatamente i 5 tipi di LSA, inviati rispettivamente dai router indicati nella seguente figura:



Tipo 1: router links advertisement
È simile a un LSP tradizionale e contiene le informazioni sui router adiacenti e sulle LAN a esso collegate; viene propagato solo all'interno dell'area e può essere generato e propagato da un backbone router sul backbone.

Tipo 2: network links advertisement
È simile a un tradizionale LSP generato per conto di una LAN da un router particolare, il Designated Router (DR) che elenca tutti i router presenti sulla LAN: il LSA viene propagato sul backbone anch'esso dai backbone router.

Tipo 3: network summary link advertisement
Un'area border router genera LSA diversi nelle aree cui è collegato: il LSA di tipo 3 è utilizzato dagli area border router per riassumere e propagare informazioni su una singola area e riguarda una sola destinazione IP; ogni area border router genera molti LSA e li mette tutti nello stesso pacchetto e li invia per ogni indirizzo IP esterno all'area e interno all'AS e sul backbone (uno per ogni indirizzo IP interno alle sue aree).

Tipo 4: AS boundary routers summary link advertisement
Serve per propagare altre informazioni di livello 2 nelle aree, come il costo dal router che ha generato il LSA, verso un AS boundary router.

Tipo 5: AS external link advertisement
Serve per propagare le informazioni di livello 2 a tutti i router dell'AS; viene generato da un AS boudary router e trasmette il costo dal router che ha generato il LSA a una destinazione esterna all'AS.

L'algoritmo OSPF

Dopo aver descritto i singoli sotto-protocolli analizziamo ora il funzionamento dell'algoritmo:

- **A** il primo passo consiste nel riconoscimento dei router vicini che si realizza con lo scambio di messaggi di "Hello";
- **B** ogni router determina il costo dei propri rami uscenti e invia questi dati istantaneamente a tutti i router della rete utilizzando la tecnica di flooding;
- **C** ogni nodo continua a monitorare il costo dei propri rami ed effettua il ricalcolo con l'algoritmo di Dijkstra (calcola l'insieme dei cammini a costo minimo – Shortest Path Tree SPT) e, nel caso individui una variazione del costo di un ramo, invia il nuovo valore a tutti i router;
- **D** a regime tutti i nodi hanno "completato" il Database Topologico o Link State Database uguale per ogni router;
- **E** utilizzando il database topologico ogni router costruisce la propria tabella di routing.

> In linea generale si può dire che il tempo di esecuzione dell'algoritmo OPSF è n log n, dove n è il numero degli archi del grafo (che rappresentano i collegamenti tra i vertici, rappresentanti i router).

Al crescere della rete corrisponde quindi un consumo sempre più elevato di memoria e CPU in ogni router: non è pertanto indicata la gestione di reti estese ma, come sappiamo, la possibilità di suddividere la rete in aree più piccole e applicare l'algoritmo su di esse permette di superare questa limitazione.
Proprio la possibilità di gestire la scalabilità delle reti ha reso OSPF molto più efficiente del protocollo RIP, basato anche esso su Link State, anche perché aumenta la sicurezza in quanto tutti gli scambi tra router OSPF sono autenticati.

> A differenza di RIP che utilizza UDP (come vedremo in seguito), i messaggi di OSPF viaggiano incapsulati direttamente in pacchetti IP: il campo Protocol nel pacchetto IP viene settato a 89 che è il codice del protocollo OSPF.



Zoom su...

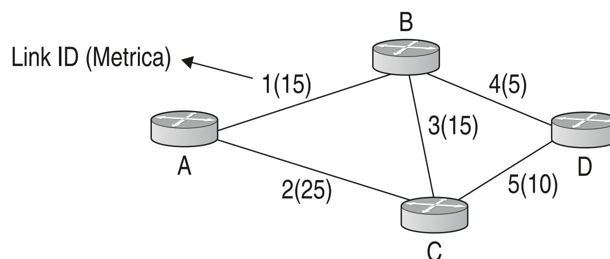
LIMITI AREA OSPF

I limiti consigliati per l'applicazione di OSPF su di un'area sono:

- non deve avere più di 50 router (è consigliato ridurne il numero in caso di link non stabili);
- un router non deve avere più di 60 neighbor;
- un router non deve stare in più di tre aree (una backbone area e una o due aree non backbone);
- il Designated Router e il Backup Designated Router devono essere dei router il più possibile scarichi.

ESEMPIO 17

Costruiamo per esempio il database topologico per la rete di figura: ▶



Al termine degli scambi di informazioni si ottiene la seguente tabella:

Database topologico

Da	A	Link	Costo
A	B	1	15
A	C	2	25
B	A	1	15
B	C	3	10
B	D	4	5
C	A	2	25
C	B	3	10
C	D	5	10
D	B	4	5
D	C	5	10

A partire da essa, ogni router calcola i percorsi a costo più basso e si costruisce la tabella di routing: di seguito è riportata quella del router A.

Routing Table del router A

Destinazione	Costo	Next hop
B	15	B
C	25	C
D	20	B

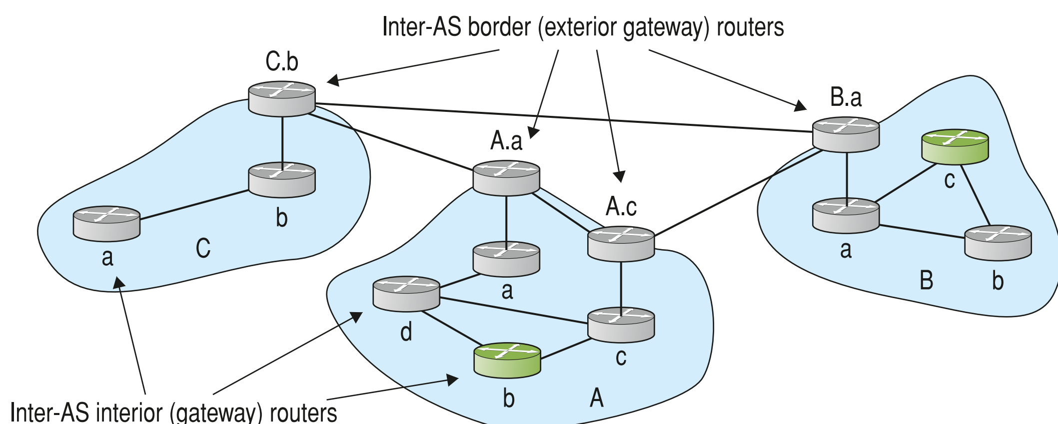
Exterior Gateway Protocol (EGP)

Gli EGP sono i protocolli che vengono utilizzati nella comunicazione i tra i router di confine: questi ruoter devono comunicare con gli altri router interni al loro AS di appartenenza e sono inoltre i responsabili del routing verso destinazioni esterne al proprio AS comunicando con gli altri gateway router.

Devono quindi eseguire sia protocolli di routing intra-AS che di routing inter-AS.

ESEMPIO 18

Vediamo un esempio di funzionamento del protocollo EGP.



Nell'esempio di figura, supponiamo che il router b appartenente al dominio A voglia trasmettere un messaggio al router c del dominio B: nella sua tabella inoltra il messaggio al router di confine A.c (router c della rete A) che ha nella sua tabella di routing le destinazioni remote e tra quelle individua il router B.a (router a delle rete B) il quale, a sua volta, conosce le destinazioni della rete B e quindi può consegnare il pacchetto al destinatario, il router c.

Exterior Gateway Protocol (EGP)

È stato il primo protocollo EGP tra AS che dà il nome alla famiglia di protocolli: risale ai primi anni ottanta (RFC 827) e oggi non è praticamente più utilizzato.

EGP è caratterizzato da tre funzionalità principali, che sono state poi mantenute nei successivi protocolli:
- neighbor acquisition: verificare se esiste un accordo per diventare vicini;
- neighbor reachability: monitorare le connessioni con i vicini;
- network reachability: scambiare informazioni sulle reti raggiungibili da ciascun vicino.

I principali problemi che hanno determinato l'abbandono dell'utilizzo di EGP a favore del BGP sono i seguenti:
- non propaga distanze tra reti, ma solo la raggiungibilità;
- presuppone un core system, non funziona con l'attuale struttura a dorsali multiple;
- non supporta load sharing né cammini alternativi;
- la convergenza può essere molto lenta;
- non supporta il policy routing;
- va in crisi con router malfunzionanti (pubblicizzano route inconsistenti).

Border Gateway Protocol (BGP)

Il BGP è di fatto lo standard nei protocolli EGP: sviluppato nel 1989 oggi è alla sua revisione 4 e utilizza la tecnica path vector, una generalizzazione del distance vector, dove ogni messaggio contiene una lista di percorsi e ogni Border Gateway comunica a tutti i vicini l'intero cammino (cioè la sequenza di AS) verso una specifica destinazione.

Il problema del routing tra AS è diverso da quello di routing interno. BGP utilizza i messaggi scambiati tra i border router per costruire un grafo di AS: è impensabile applicare le tecniche utilizzate per la ricerca dei percorsi ottimi in metriche che permettano di individuare i cammini intra-AS; inoltre l'instradamento deve tener conto anche di eventuali preferenze per alcuni AS rispetto ad altri (accordi commerciali, prestazioni ecc.) e quindi i parametri sui quali effettuare la scelta nei protocolli EGP sono molteplici e spesso differenti.

Nella scelta dei percorsi è necessario conoscere l'intero percorso verso la destinazione quindi il DV non va bene perché non consente la conoscenza dell'intero percorso e neppure il LS perché occorrerebbe costruire informazioni topologiche sull'intera rete mondiale.
Viene utilizzato il path vector: il BGP è un protocollo simile al distance vector, però nei DV inviati dai nodi non è indicata una "distanza dalla destinazione", ma l'intero percorso verso la destinazione:

Rete	Router successivo	Percorso
N01	R01	AS2,AS5,AS7,AS12
N02	R07	AS4,AS13,AS6,AS9
N03	R09	AS11,AS12,AS8,AS6
…	…	…

In realtà un messaggio di path vector che si scambiano due EG vicini non contiene solo un percorso ma una sequenza di "attributi" che si distinguono in attributi obbligatori, che devono essere interpretati da tutte le implementazioni di BGP, e facoltativi.

La struttura del percorso che viene scambiata tra router è la seguente:

Network IP Address + Netmask + Path attribute

Il campo Path attribute è costituito da numerosi attributi, dove il più importante è la lista degli AS (AS_PATH): ciascun router memorizza la lista degli AS attraversati e poi invia un aggiornamento agli altri vicini inserendo il proprio AS di appartenenza.
Tra gli attributi obbligatori i più importanti sono:
- ORIGIN: protocollo IGP da cui proviene l'informazione (per esempio OSPF, RIP, IGRP);
- NEXT_HOP: prossimo router.



Zoom su...

CARATTERISTICHE DEGLI ATTRIBUTI

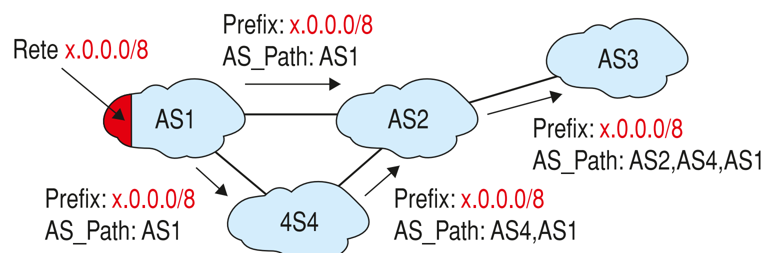
Le caratteristiche di ciascun attributo sono codificate in 2 byte:

0	1	2	3	4–7	8–15
O	T	P	E	0	Attribute Type Code

dove i campi indicano rispettivamente:
- O (well-known/optional): indica se l'attributo è opzionale (O=1) oppure se è ben conosciuto;
- T (transitive/local):
- transitive: (T=1) è un parametro che può essere aggiornato o interpretato da altri router e deve quindi essere trasmesso ai router successivi;
- local: (T=0) parametro che assume un significato solo all'interno di un certo AS; non deve essere propagato;
- P (partial): settato se qualche router sul percorso non è stato in grado di comprendere il significato di quel particolare attributo (per esempio uno opzionale); significa che l'attributo è stato solo parzialmente valutato;
- E (length): indica se la lunghezza del campo LENGTH è codificata con 1 ottetto (E = 0) oppure con 2.

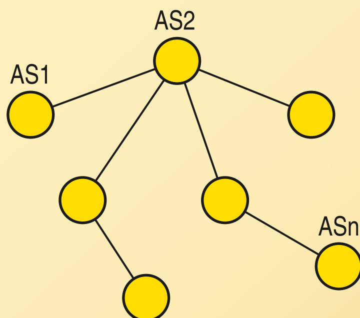
ESEMPIO 19

Vediamo un esempio di funzionamento nella rete riportata in figura, dove propaghiamo il percorso per raggiungere la rete x.0.0.0/8 appartenente al AS1.



AS1 comunica a AS2 e ad AS4 il percorso per raggiungere x.0.0.0/8 e AS4 successivamente lo integra e lo ritrasmette ad AS2 (AS_PATH: AS4,AS1).
AS2 può scegliere se raggiungere la rete X.0.0.0/8 attraverso AS4 o AS1: supponiamo che scelga AS4 per meri motivi economici e quindi inoltra ad AS3 un path vector che riporta il percorso da lui scelto per raggiungere tale rete: AS2-AS4-AS1.

Ogni router si costruisce un albero, l'AS path tree, che per l'AS2 del nostro esempio è il seguente:



BGP consente a ciascuna sottorete di comunicare la propria esistenza al resto di Internet: offre infatti gli strumenti per propagare le informazioni di raggiungibilità a tutti i router interni dei diversi AS.

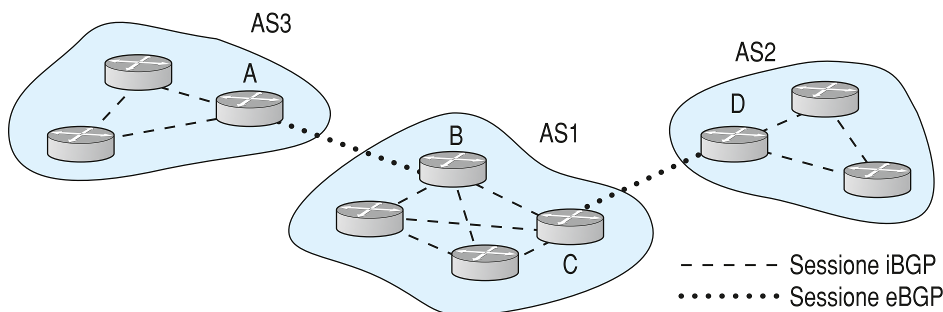
Lo scambio di messaggi BGP è supportato da connessioni TCP sulla porta Porta 179: la comunicazione è quindi affidabile e non sono necessari meccanismi di ritrasmissione e recupero d'errore.
Quando due router stabiliscono tra loro connessioni TCP per lo scambio di messaggi BGP si parla di sessione BGP e peer BGP che, a seconda del tipo di router, prende anche il nome di;
- iBGP: sessioni interne, solo tra router interni;
- eBGP: sessioni esterne, solo tra router di confine.



BGP SPEAKER
Un router che supporta il protocollo BGP è chiamato BGP speaker, e non necessariamente coincide con un AS border router.

ESEMPIO 20

Nella semplice rete di figura sono indicate le sessioni iBGP ed eBGP.



Per esempio AS3 invia ad AS1 in una sessione eBGP tra i gateway A e B la lista delle sottoreti raggiungibili specificandone il prefisso (CIDR): il router B utilizza le proprie sessioni iBGP per propagare l'informazione agli altri router del sistema autonomo. In modo analogo anche AS1 e AS2 si scambiano informazioni sulla raggiungibilità dei prefissi attraverso i propri gateway C e D.



Zoom su...

TIPI DI MESSAGGIO

Possiamo classificare i messaggi che periodicamente si scambiano i router in quattro gruppi:

- OPEN: servono per inizializzare la connessione tra router, autenticando il mittente;
- UPDATE: sono messaggi che aggiornamento delle informazioni di raggiungibilità:
  - annuncio di un nuovo cammino (PATH);
  - eliminazione di un cammino preesistente (WITHDRAWN);
- NOTIFICATION: vengono utilizzati come risposta a un messaggio errato oppure per chiudere una connessione;
- KEEPALIVE: permettono di verificare lo stato di un altro router mantenendo attiva la connessione anche se due router non si scambiano messaggi di UPDATE.



Summary of Technology of Distributed Routing

Link State

- Topology information is flooded within the routing domain
- Best end-to-end paths are computed locally at each router.
- Best end-to-end paths determine next-hops.
- Based on minimizing some notion of distance
- Works only if policy is shared and uniform
- Examples: OSPF, IS-IS

Vectoring

- Each router knows little about network topology
- Only best next-hops are chosen by each router for each destination network.
- Best end-to-end paths result from composition of all next-hop choices
- Does not require any notion of distance
- Does not require uniform policies at all routers
- Examples: RIP, BGP

	Link State	Vectoring
IGP	OSPF IS-IS	RIP
EGP		BGP

(courtesy of Timothy G. Griffin Intel Research, Cambridge UK)

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Con ASBR si intende:
a) autonomous system Boundary Router
b) autonomous system Border Router
c) autonomous system Backbone Router
d) area 0 system Boundary Router
e) area 0 system Border Router

2 Quali tra i seguenti protocolli sono utilizzati dai IR?
a) RIP
b) OSPF
c) EGP
d) IGRP
e) BGP
f) EIGRP
g) Integrated IS-IS

3 Quali tra i seguenti protocolli utilizzano il Distance Vector?
a) RIP
b) OSPF
c) IGRP
d) EIGRP
e) Integrated IS-IS

4 Il formato del messaggio RIP v1 ha i seguenti campi:
a) Command
b) Version
c) 25 address
d) Metric
e) Netmask

5 OSPF è costituito da 3 "sotto-protocolli":
a) protocollo Hello, Exchange, Flooding
b) protocollo Hello, Exchange, Update
c) protocollo Update, Exchange, Flooding
d) protocollo Hello, Exchange, Flooding

6 Il campo *type* specifica il tipo di messaggio OSPF che può essere (indica quello errato):
a) hello
b) database update
c) link status request
d) link status update
e) link status acknowledgement

7 Il record "Database Description Packet" viene utilizzato:
a) nel protocollo Exchange
b) nel protocollo Hello
c) nel protocollo Flooding
d) in tutti i protocolli precedenti

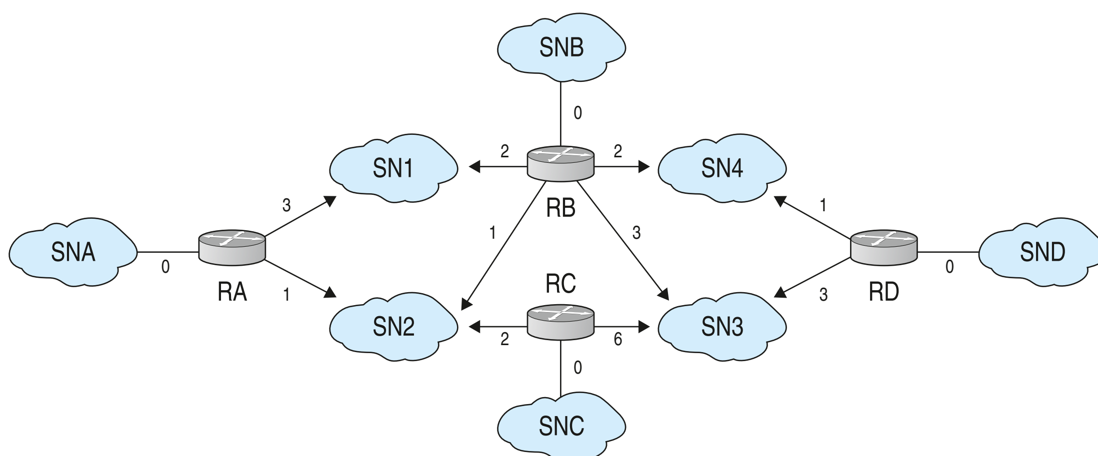
Test vero/falso

1 Nel routing gerarchico l'internetwork viene suddivisa in varie regioni. V F
2 Gli autonomous system (AS) sono dei raggruppamenti logici di computer. V F
3 I router interni non possono comunicare con i router di frontiera. V F
4 Con router di confine ABR si intende Area Boundary Router. V F
5 Ogni AS è univocamente identificato tramite un nome composto da 16 bit. V F
6 Un AS con un solo router di confine prende il nome di stub. V F
7 I router multi-homed non-transit sono generalmente di grandi corporate. V F
8 Il RIPv2 modifica il record che viene trasmesso aggiungendo l'indirizzo IP. V F
9 L'OSPF ha una maggiore velocità di convergenza rispetto al RIP. V F

Verifichiamo le competenze

Esercizi

1 In figura è riportato un sistema autonomo formato da sottoreti e da router dove sono indicati i costi dei singoli rami come sono visti dai diversi router.
Il protocollo IGP utilizzato è il protocollo OSPF con metrica coincidente con i costi dei rami.



Le sottoreti hanno gli indirizzi mostrati nella seguente tabella:

Sottorete	Net_id	Sottorete	Net_id
SNA	215.0.0	SN1	195.0.0
SNB	218.0.0	SN2	200.0.0
SNC	220.0.0	SN3	205.0.0
SND	222.0.0	SN4	210.0.0

Ogni router viene identificato con il proprio nome come valore di host_id=nome: per esempio se l'indirizzo di sottorete SN2 è 150.0.0, RC su SN2 viene visto come 150.0.0.C.

1) Completa i messaggi LSP inviati dal router RA sulle proprie interfacce verso tutti gli altri router della rete, in seguito alla scoperta dei propri vicini.

A	Link to	215.0.0
	Metric	0

A	Link to	
	Metric	

A	Link to	
	Metric	

2) Completa la tabella di instradamento (routing table) del router RA.

Sottorete	Routing (next hop)	Distanza	Sottorete	Routing (next hop)	Distanza
SNA	Local	0	SN1		
SNB	200.0.0	1	SN2		
SNC			SN3		
SND			SN4		

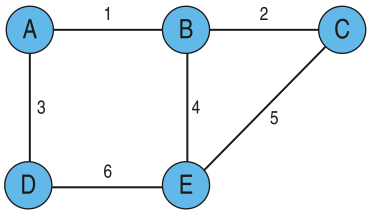
3) Supponiamo che il peso del ramo uscente dal router RC verso la rete SN3 divenga uguale a 1. Completa il seguente messaggio che il protocollo RIP emette verso i suoi adiacenti per descrivere questo aggiornamento.

Sottorete	Metrica	Sottorete	Metrica
SNA		SN1	
SNB		SN2	
SNC		SN3	
SND		SN4	

4) Riscrivi la tabella di instradamento del router RA a seguito della variazione indicata al punto precedente.

Sottorete	Routing (next hop)	Distanza	Sottorete	Routing (next hop)	Distanza
SNA	Local	0	SN1		
SNB			SN2		
SNC			SN3		
SND			SN4		

2 Scrivi il database topologico secondo il protocollo OSFP della rete riportata in figura dove i link numerati sono di peso unitario e da esso ricava quindi la tabella di routing per tutti i router.



Database topologico (link bidirezionali)

Da	A	Link	Distanza
A	B		
A	D		
B	C		
B	E		
D	E		
E	C		

Routing table

A	Destinazione	A	B	C	D	E
	Distanza					
	Link	local				

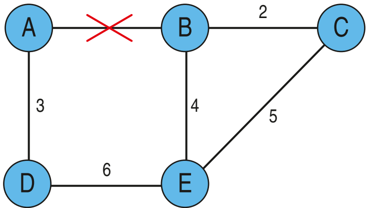
B	Destinazione	A	B	C	D	E
	Distanza					
	Link		local			

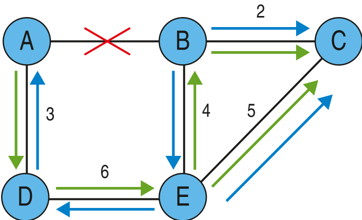
C					
Destinazione	A	B	C	D	E
Distanza					
Link			local		

D					
Destinazione	A	B	C	D	E
Distanza					
Link				local	

E					
Destinazione	A	B	C	D	E
Distanza					
Link					local

3 Scrivi il database topologico secondo il protocollo OSFP della rete dell'esercizio precedente in caso di rottura del link tra il router A e il router B.





A		
Link to	1
Metric	Inf

B		
Link to	1
Metric	Inf

Database topologico (link bidirezionali)

Da	A	Link	Distanza
A	B		
A	D		
B	C		
B	E		
D	E		
E	C		

Routing table

A					
Destinazione	A	B	C	D	E
Distanza					
Link	local				

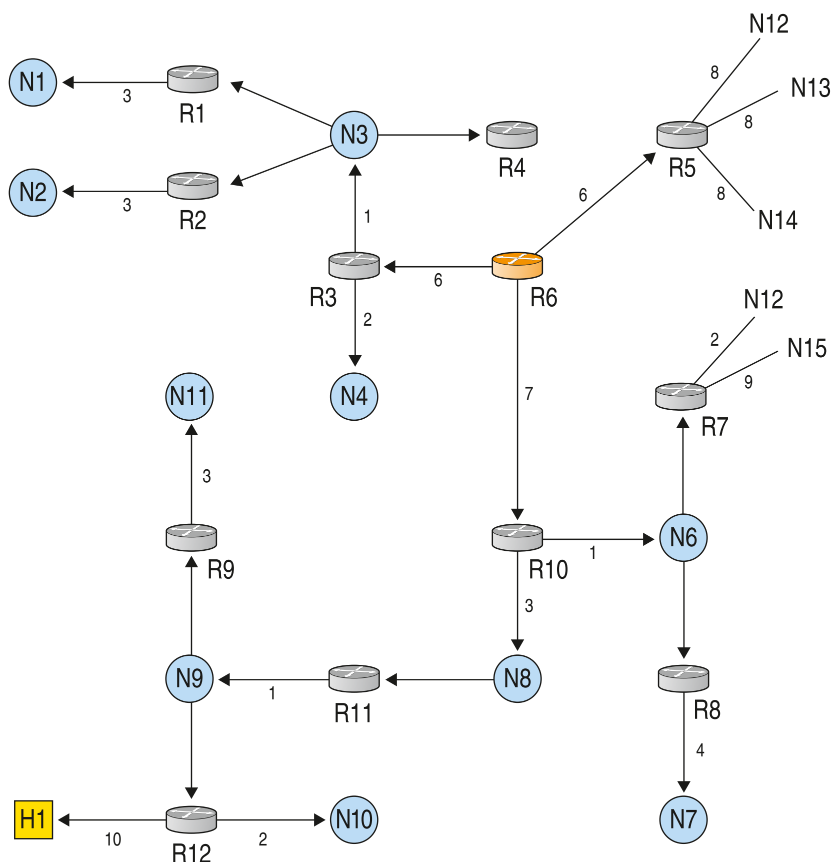
B	Destinazione	A	B	C	D	E
	Distanza					
	Link		local			

C	Destinazione	A	B	C	D	E
	Distanza					
	Link			local		

D	Destinazione	A	B	C	D	E
	Distanza					
	Link				local	

E	Destinazione	A	B	C	D	E
	Distanza					
	Link					local

4 In riferimento alla rete di figura, scrivi la tabella di routing per il nodo R6 utilizzando il protocollo OSFP



R6

Destinazione	Next Hop	Distanza
N1	R3	
N2	R3	
N3	R3	
N4	R3	
R1	R3	
N6	R10	
N7	R10	
N8	R10	
N9	R10	
N10	R10	
N11	R10	
H1	R10	
R5	R5	
R7	R10	
N12	R10	
N13	R5	
N14	R5	
N15	R10	

Esercitazioni di laboratorio 1

ROTTE STATICHE: IL COMANDO ROUTE

Il comando ROUTE è un comando con il quale è possibile impostare una rotta statica a una determinata mask e un determinato IP di un host di una rete locale.
Digitando semplicemente route al prompt dei comandi vengono elencate tutte le opzioni disponibili.

ROUTE [-f] [-p] comando [destinazione] [MASK netmask] [gateway] [METRIC passi] [interfaccia IF]

Dove il parametro comando può essere (con SO windows):
- PRINT: visualizza una route;
- ADD: aggiunge una route;
- DELETE: elimina una route;
- CHANGE: modifica una route.

Con i comandi PRINT/DELETE è possibile utilizzare caratteri jolly * in destinazione o gateway.

I parametri opzionali hanno le seguenti funzioni:
- -f: cancella le tabelle di routing di tutte le voci gateway;
- -p: quando si usa il comando ADD mantiene la route a ogni avvio del sistema;
- destinazione: specifica l'host;
- MASK netmask: valore della maschera per la subnet da associare alla voce di route;
- gateway: specifica il gateway;
- METRIC passi: specifica il numero di passi/costo per la destinazione;
- interfaccia IF: numero di interfaccia per la route specificata (nel caso di presenza di più schede).

Per esempio, per visualizzare la tabella di routing dell'host si digita

```
route print                        (Windows)
route –n                           (Linux/Unix)
```

Invece il comando

```
route –p add DEST mask NETMASK GATEWAY
```

aggiunge alla tabella di routing Windows una entry permanente relativa alla destinazione DEST indicandone la NETMASK e il GATEWAY attraverso il quale raggiungerla.

Il comando

```
route ADD 157.0.0.0 MASK 255.0.0.0 157.55.80.1 METRIC 3 IF 2
```

aggiunge un percorso con
- destinazione 157.0.0.0
- mask 255.0.0.0
- gateway 157.55.80.1
- metric 3
- interfaccia 2



Prova adesso!

Visualizza la tabella presente sul tuo host, che sarà simile alla seguente:

```
C:\WINDOWS\system32\CMD.exe

C:\>ROUTE PRINT
===========================================================================
Elenco interfacce
0x1 ........................... MS TCP Loopback interface
0x2 ...90 fb a6 ed f8 3a ...... Realtek PCIe GBE Family Controller - Miniport de
ll'Utilitó di pianificazione pacchetti
===========================================================================
===========================================================================
Route attive:
Indirizzo rete             Mask          Gateway       Interfac.  Metric
          0.0.0.0          0.0.0.0    192.168.1.254    192.168.1.2     10
        127.0.0.0        255.0.0.0        127.0.0.1      127.0.0.1      1
      192.168.1.0    255.255.255.0      192.168.1.2    192.168.1.2     10
      192.168.1.0  255.255.255.255      192.168.1.2    192.168.1.2      1
      192.168.1.2  255.255.255.255        127.0.0.1      127.0.0.1     10
    192.168.1.255  255.255.255.255      192.168.1.2    192.168.1.2     10
        224.0.0.0        240.0.0.0      192.168.1.2    192.168.1.2     10
  255.255.255.255  255.255.255.255      192.168.1.2    192.168.1.2      1
Gateway predefinito:     192.168.1.254
===========================================================================
Route permanenti:
  Indirizzo rete           Mask     Indir. gateway   Metric
      192.168.1.0  255.255.255.255      192.168.1.2        1

C:\>
```

Dalla tabella possiamo individuare immediatamente l'indirizzo IP dell'host ricercando nella colonna di Gateway l'indirizzo di loopback: in questo caso nella quarta riga individuiamo 192.168.

Sempre analizzando la colonna di Gateway osserviamo che abbiamo solo 3 possibili destinazioni:
- Gateway = IP locale: in questo caso avviene la consegna diretta;
- Gateway = loopback: in questo caso il pacchetto non viene inoltrato ma consegnato agli strati superiori della pila protocollare;
- negli altri casi avviene la consegna indiretta tramite il default gateway.

Vediamo un esempio di instradamento utilizzando la nostra tabella: ricordiamo che il controllo viene effettuato a partire dalla riga che presenta una netmask con un numero maggiore di bit a uno, in modo da dare priorità alle route più specifiche: prima host, poi reti piccole, poi reti grandi (longest-prefix match).

1 Datagramma con IP dest. = 192.168

192.168.001.002
255.255.255.255

192.168.001.002 == 192.168.001.002

La riga 4 è quella giusta (host specific).

2 Datagramma con IP dest. = 192.168.1.22

192.168.001.022
255.255.255.255

192.168.001.022 != 192.168.001.002
192.168.001.022
255.255.255.000

192.168.001.000 == 192.168.001.000

La riga 3 è quella giusta (network specific).

3 Datagramma con IP dest. = 80.48.15.170

080.048.015.170
255.255.255.255

080.048.015.170 != 192.168.001.002

080.048.015.170
255.255.255.000

080.048.015.000 != 192.168.001.000

080.048.015.170
000.000.000.000

000.000.000.000 == 000.000.000.000

La riga 1 è quella giusta (default gateway).

Aggiunta di una entry

Modifichiamo ora la nostra tabella aggiungendo una nuova riga: per prima cosa individuiamo un **PC** sulla rete, possibilmente di un'aula o in un laboratorio diverso da quello dell'host mittente in modo da avere almeno un router che effettui il gateway.

Individuato l'indirizzo **IP** del destinatario (per esempio **192.168.x.x**) individuiamo il percorso per raggiungerlo mediante il comando **TRACERT**.

```
C:\>tracert 192.168.1.2

Rilevazione instradamento verso nome-c5eb64fc20 [192.168.1.2]
su un massimo di 30 punti di passaggio:

  1    74 ms     1 ms     1 ms  192.168.1.200
  2     1 ms     1 ms     1 ms  192.168.1.2

Trace complete.
```

Si può osservare come il pacchetto giunga a destinazione attraverso un router intermedio: aggiungiamo alla nostra tabella l'indirizzo di tale router instradando quindi direttamente il pacchetto facendo così in modo da evitare che in futuro passi dal primo router:

```
C:\>route -p add 192.168.1.0 mask 255.255.255.0 192.168.1.200
```

Visualizziamo ora la tabella del nostro host:

```
C:\WINDOWS\system32\cmd.exe
C:\>
C:\>route print
===========================================================================
Elenco interfacce
0x1 ........................... MS TCP Loopback interface
0x2 ...00 19 99 62 58 3b ...... Broadcom NetLink (TM) Gigabit Ethernet - Minipor
t dell'Utilitó di pianificazione pacchetti
===========================================================================
===2.100===================================================================
Route attive:
Indirizzo rete             Mask          Gateway       Interfac.  Metric
          0.0.0.0          0.0.0.0      192.168.2.2   192.168.2.190     20
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1      1
      192.168.2.0    255.255.255.0    192.168.2.190   192.168.2.190     20
      192.168.1.0    255.255.255.0    192.168.1.200   192.168.1.200     20
    192.168.1.100  255.255.255.255    192.168.1.254   192.168.2.190     20
    192.168.2.255  255.255.255.255    192.168.2.190   192.168.2.190     20
        224.0.0.0        240.0.0.0    192.168.2.190   192.168.2.190     20
  255.255.255.255  255.255.255.255    192.168.2.190   192.168.2.190      1
Gateway predefinito:       192.168.2.2
===========================================================================
Route permanenti:
  Nessuno
```

Proviamo ora a ripetere la stessa operazione con **TRACERT** e otteniamo

```
C:\>tracert 192.168.1.2

Rilevazione instradamento verso nome-c5eb64fc20 [192.168.1.2]
su un massimo di 30 punti di passaggio:

  1     1 ms     1 ms     1 ms  nome-c5eb64fc20 [192.168.1.2]

Rilevazione completata.
```

Il nostro pacchetto è stato consegnato direttamente.

Esercitazioni di laboratorio 2

CONNESSIONE DI RETI MEDIANTE ROUTER

Considera la topologia di rete IP costituita da 2 router, 2 switch e 5 host che sono tra loro organizzati come due reti, una di classe C e l'altra di classe B.

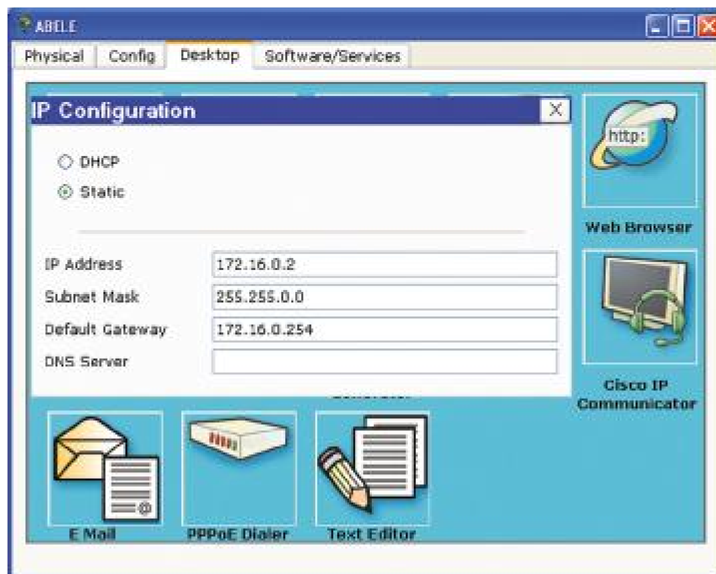
Rete LAN A Classe C
PC 192.168.0.X con X = 1....253
Gateway 192.168.0.254

Inserisci per esempio tre PC con i seguenti indirizzi:
- QUI: 192.168.0.1
- QUO: 192.168.0.2
- QUA: 192.168.0.3

Rete LAN B Classe B
PC 172.16.0.X con X = 1....65533
Gateway 172.16.255.254

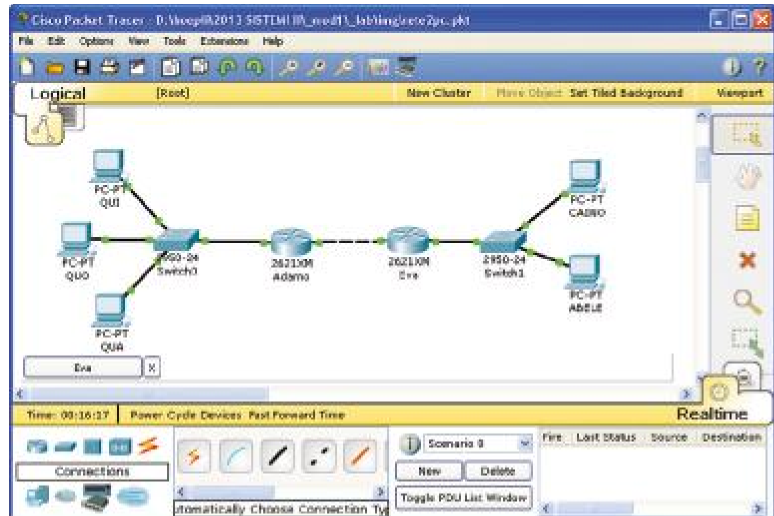
Inserisci per esempio due PC con i seguenti indirizzi:
- CAINO: 172.16.0.1
- ABELE: 172.16.0.2

utilizzando la scheda Desktop del PC come riportato nella figura: ▶



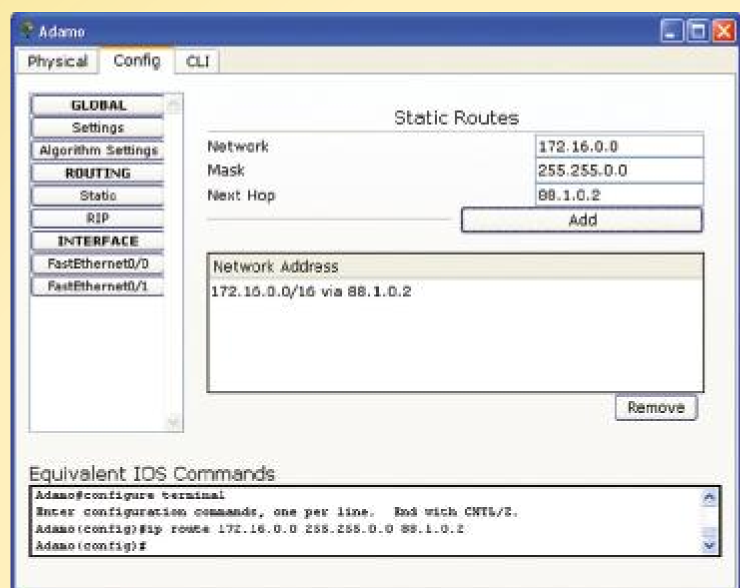
Realizza la rete utilizzando due router 2621XM come nello schema riportato in figura, assegnando ai due router i seguenti indirizzi:

- Adamo: 88.1.0.1/16
- Eva: 88.1.0.2/16

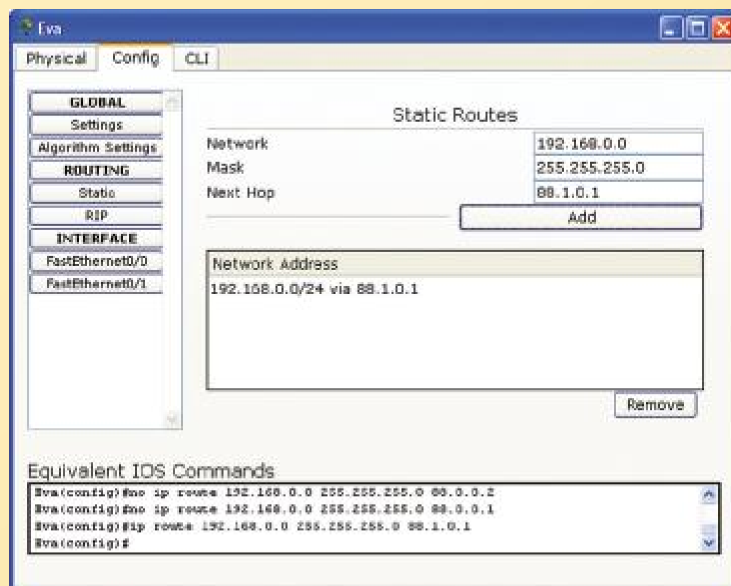


1. Assicurati che le interfacce del router siano ON osservando i "led" ai capi dei link che le collegano ai relativi switch.
2. Imposta su tutti i PC e sui server l'indirizzo corretto del default gateway (sarà l'indirizzo IP dell'interfaccia del router sulla propria LAN).
3. Invia un messaggio dal PC QUI della LAN A al PC QUO della LAN A e nella Event list osserva solo i pacchetti ICMP.
4. Invia un messaggio dal PC QUA delle LAN A al PC ADAMO della LAN B.
5. Analizza lo scambio di pacchetti e descrivi le differenze tra le due diverse situazioni.
6. Al termine della simulazione visualizza la tabella di ARP con il command prompt del PC CAINO del router ADAMO.

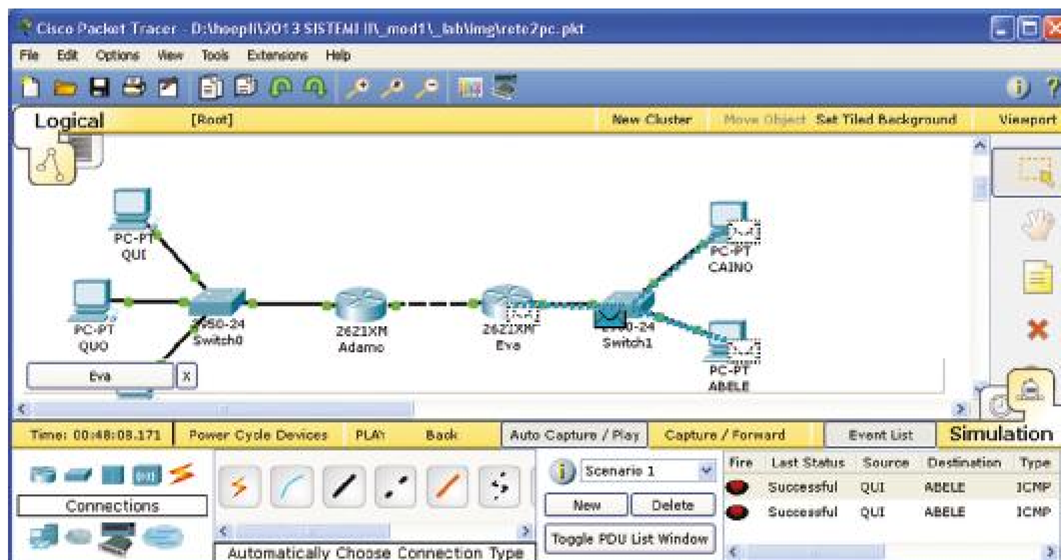
Molto probabilmente la mancata consegna del messaggio al PC ADAMO è dovuta alla mancanza di configurazione del percorso, come vedremo in seguito: per poter far comunicare le due reti è necessario aggiungere una riga nella tabella del router ADAMO come quella di figura ▶



e una riga nella tabella del router EVA come nella figura seguente.



Dopo aver configurato "correttamente" la route statica è possibile effettuare la consegna tra PC appartenenti a reti diverse, come si può vedere nella successiva figura:



(Il file di questo esercizio è salvato col nome rete2pc.pkt ed è scaricabile dal sito dedicato a questo volume).

Esercitazioni di laboratorio 3

ROTTE STATICHE: CONFIGURAZIONE E GESTIONE

La definizione delle rotte statiche viene realizzata manualmente dall'amministratore di rete impostando le entry nella tabella di routing: sono utilizzate per piccole reti a causa della scarsa scalabilità.
Per impostare una rotta statica è necessario specificare:
- Indirizzo IP della rete di destinazione;
- Netmask associata alla rete di destinazione;
- Indirizzo IP del next-hop (oppure il nome dell'interfaccia per forzare la consegna locale).

Il comando CLI che permette di inserire in un router una rotta statica è il seguente:

```
Router(config)#ip route DestPrefix DestNetmask NextHop/Iface
Router(config)#
```

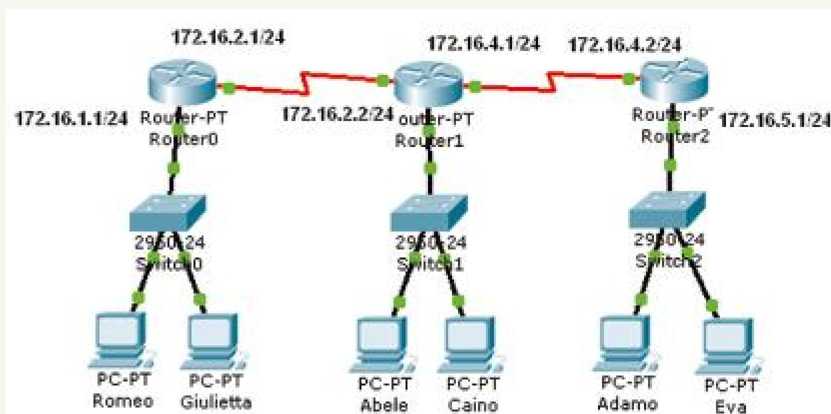
Il comando che permette di rimuovere in un router una rotta statica è il seguente:

```
Router(config)#no ip route DestPrefix DestNetmask NextHop/Iface
Router(config)#
```



Prova adesso!

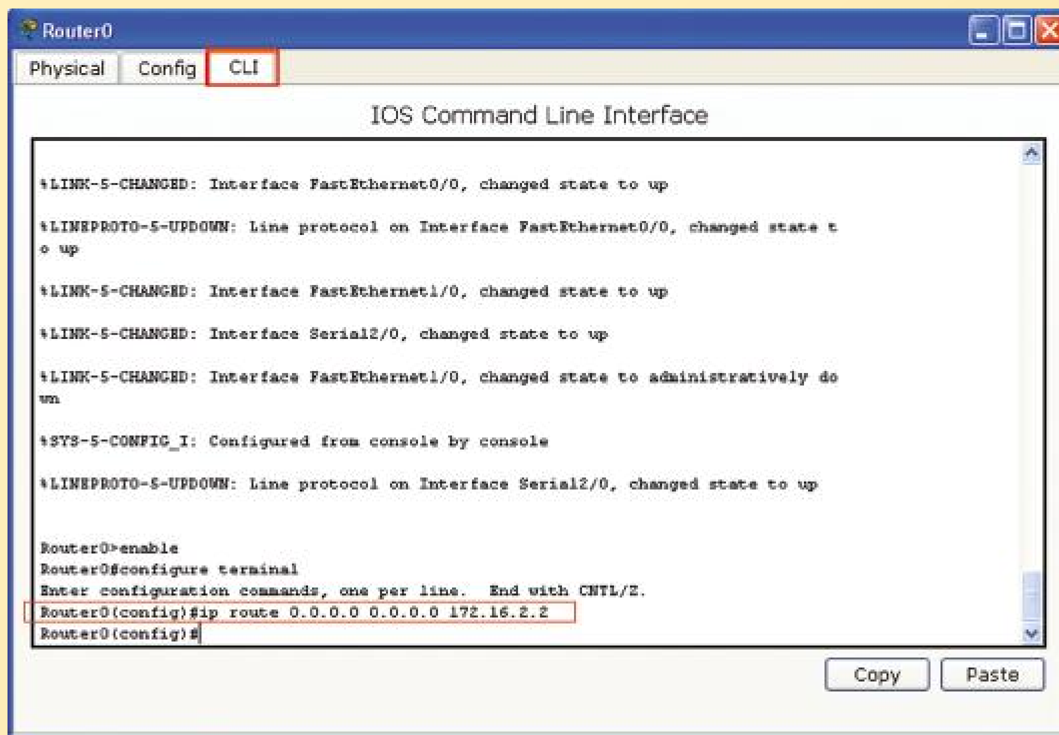
Realizza con Packet Tracer la rete riportata nella seguente figura:



e configura i singoli router con i seguenti comandi:

Router0(config)#ip route 0.0.0.0 0.0.0.0 172.16.2.2
Router1(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.1
Router1(config)#ip route 172.16.5.0 255.255.255.0 172.16.4.2
Router2(config)#ip route 0.0.0.0 0.0.0.0 172.16.4.1

L'inserimento delle route deve essere fatto mediante la finestra presente nella scheda CLI, cioè



Ping e Traceroute su router

In PacketTracer, la sintassi di entrambi i comandi differisce tra PC e router:

ping

```
Router#ping IP_ADDRESS
PC>ping [-n COUNT] IP_ADDRESS
```

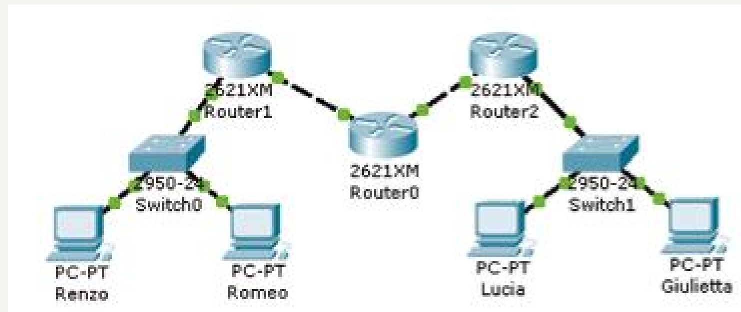
traceroute

```
Router#traceroute IP_ADDRESS
PC>tracert IP_ADDRESS
```



Prova adesso!

1 Crea la rete mostrata in figura



assegnando alle interfacce di rete dei router e degli host degli indirizzi IP appartenenti alle seguenti reti:

- `192.168.1.0/24` : *Renzo, Romeo* e *Router1 Fa0/1*
- `192.168.2.0/24` : *Router1 Fa0/0* e *Router0 Fa0/0*
- `192.168.3.0/24` : *Router2 Fa0/1* e *Router2 Fa0/0*
- `192.168.4.0/24` : *Router1 Fa0/, Lucia e Giulietta*

2 Imposta le rotte statiche opportune su tutti i nodi della rete e verifica la completa connettività della rete col comando traceroute da Renzo a Lucia.

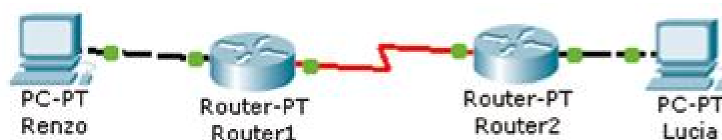
3 In modalità simulazione, analizza i pacchetti generati dal comando traceroute.
Che messaggi usa e in che modo?

Il codice con la soluzione dei presenti esercizi è nei file rete3pc.pkt e rete4pc.pkt.

Esercitazioni di laboratorio 4

ROTTE STATICHE: COLLEGAMENTO SERIALE

Si vuole far "pingare" i due PC, rappresentati nello schema di figura, attraverso due router collegati fra loro attraverso un link seriale.



Gli indirizzi di configurazione sono i seguenti
- PC Renzo: Indirizzo IP 192.168.100.2 /24 default gateway 192.168.100.1.
- PC Lucia: Indirizzo IP 192.168.200.2 /24 default gateway 192.168.200.1.

Router 1
Interfaccia FastEthernet 0/0 IP address : 192.168.100.1 /24
Interfaccia Serial 2/0 IP address: 192.168.1.1 /24.

Router 2
Interfaccia FastEthernet 0/0 IP address of 192.168.200.1 /24
Interfaccia Serial 2/0 IP address 192.168.1.2 /24.

Tra il Router1 e il Router2 abbiamo un collegamento seriale con velocità di 64K.

È richiesto di:
- configurare gli IP e i default gateway dei PC;
- configurare gli apparati dando gli hostname e impostando le rotte statiche, assicurandosi la raggiungibilità di ogni apparato tramite i comandi ping e traceroute.

Per un approfondimento delle modalità di configurazione del link seriale si rimanda al documento in inglese, riportato in allegato dal link http://www.jlsnet.co.uk/index.php?page=cc_wan, che spiega in modo chiaro le varie tipologie di connessione e configurazione.

Il codice con la soluzione del presente esercizio è nel file rete2pcSerial.



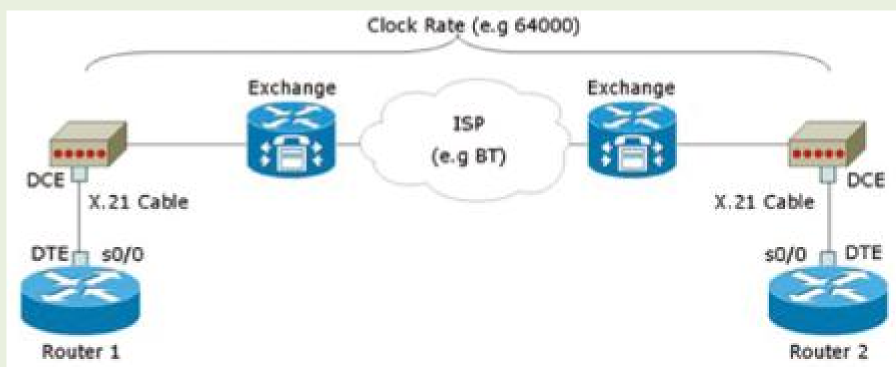
Zoom su...

CONNESSIONE TRA DTE E DCE

Ogni interfaccia seriale del router viene denominata DTE (data terminal equipment); questa viene collegata, tramite un cavo di tipo DTE/DCE (ve ne sono di diversi tipi ma il più utilizzato è il cavo denominato V.35) a un dispositivo, di solito fornito da ISP, denominato DCE (data communications equipment).

Il DCE può essere:

1. un modem analogico nel caso che la connettività fornita da ISP si basi su linee che utilizzano un segnale di tipo analogico;
2. un dispositivo CSU/CDU (ossia un terminal adapter digitale) nel caso di linee di comunicazione basate su segnale digitale.



Il DCE ha il compito di ricevere il segnale digitale trasmesso da DTE (ossia dall'interfaccia seriale del router) e di trasformarlo in un segnale (analogico o digitale) che possa essere veicolato sulla linea di comunicazione fornita da ISP.

Se in laboratorio si vuole testare il funzionamento di una comunicazione basata su link seriale si può sostituire i due CSU/DSU e la relativa linea di interconnessione con un apposito cavo incrociato DTE/DCE crossover cable come quello riportato in figura.

Le due estremità di questo cavo vengono denominate rispettivamente DTE e DCE.



Per vedere se all'interfaccia seriale del router R1 è stata collegata l'estremità di tipo DTE o DCE si utilizza il comando:

```
R1#show controllers S0/0
```

Nel nostro esercizio abbiamo:

Router1

Physical | Config | CLI

IOS Command Line Interface

```
Router0#show controllers Serial2/0
Interface Serial2/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 64000
idb at 0x81081AC4, driver data structure at 0x81084AC0
SCC Registers:
General [GSMR]=0x2:0x00000000, Protocol-specific [PSMR]=0x8
Events [SCCE]=0x0000, Mask [SCCM]=0x0000, Status [SCCS]=0x00
Transmit on Demand [TODR]=0x0, Data Sync [DSR]=0x7E7E
Interrupt Registers:
Config [CICR]=0x00367F80, Pending [CIPR]=0x0000C000
Mask   [CIMR]=0x00200000, In-srv  [CISR]=0x00000000
Command register [CR]=0x580
Port A [PADIR]=0x1030, [PAPAR]=0xFFFF
       [PAODR]=0x0010, [PADAT]=0xCBFF
Port B [PBDIR]=0x09C0F, [PBPAR]=0x0800E
       [PBODR]=0x00000, [PBDAT]=0x3FFFD
Port C [PCDIR]=0x00C, [PCPAR]=0x200
       [PCSO]=0xC20,  [PCDAT]=0xDF2, [PCINT]=0x00F
Receive Ring
        rmd(68012830): status 9000 length 60C address 3B6DAC4
        rmd(68012838): status B000 length 60C address 3B6D444
Transmit Ring
```

Copy | Paste

Nel nostro esempio il clock è già settato a 64.000 e se volessimo modificarlo si può utilizzare il comando:

```
R1(config-if)#clock rate 64000
```



Prova adesso!

Configura manualmente il router 2, come di seguito descritto:

```
...
Router(config)#hostname Router2
...
Router2(config)#interface F0/0
Router2(configif)#ip address 192.168.200.1 255.255.255.0
...
Router2(config)#interface Serial2/0
Router2(config-if)#ip address 192.168.1.2 255.255.255.0
Router2(config-if)#clock rate 64000
...
Router2(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.1 //rotta di default

Router2(configif)#exit
```

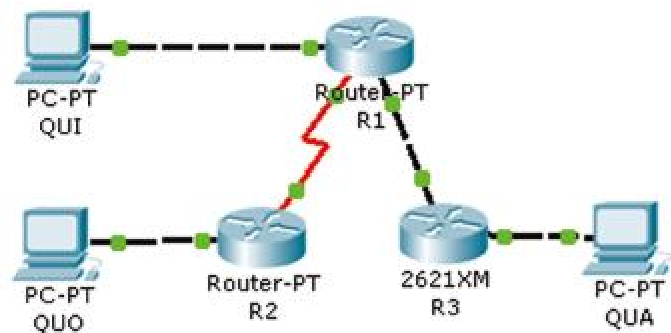
Esercitazioni di laboratorio 5

ROTTE STATICHE: COLLEGAMENTO SERIALE E ETH

Si vuole far pingare i due PC, rappresentati nello schema di figura, attraverso tre router:

- Router R1 – Generico con Serial2 DCE
- Router R2 – Generico con Serial2 DCE
- Router R3 – Serie 2600 – modello 2621XM

Il link seriale deve avere velocità di 64K.



Gli indirizzi di configurazione sono i seguenti:

A per i ruoter

```
R1(configif)#ip address 10.1.1.1 255.255.255.0
R1(configif)#ip address 12.5.10.1 255.255.255.0
R2(configif)#ip address 10.1.1.2 255.255.255.0
R3(configif)#ip address 12.5.10.2 255.255.255.0
```

B per gli host

- QUI: 192.168.200.2/24
- QUO: 192.168.100.2/24
- QUA: 150.10.0.1 /16



Prova adesso!

Configura manualmente gli apparati dando gli hostname, inserendo le rotte statiche e assicurando la raggiungibilità di ogni apparato tramite i comandi ping e traceroute.
Confronta la tua soluzione con quella presente nel file rete3R.pkt.

ESERCITAZIONI DI LABORATORIO 6

PROTOCOLLO RIP (ROUTING INFORMATION PROTOCOL)

Premessa

La prima versione del protocollo RIP, progettata a Berkley nel 1982, usa l'algoritmo di Bellman-Ford per il calcolo dei percorsi minimi e come metrica usa l'hop-count e gestisce al massimo 16 hop: una rete con distanza 16 hop è considerata irraggiungibile.
I messaggi RIP vengono inviati mediamente ogni 30 secondi:
- una rotta è considerata inutilizzabile se non viene aggiornata da 180 secondi;
- una rotta inutilizzabile per 240 secondi viene eliminata.

Comandi di diagnostica per il routing e RIP

I comandi fondamentali per poter utilizzare i router con il protocollo RIP e per effettuare la diagnostica della rete sono i seguenti:

- per configurare in un router il protocollo RIP si utilizza:

```
Router(config)#router rip
Router(config-router)#
```

- per specificare la versione di RIP da usare il comando è:

```
Router(config-router)#version N
```

- per definire su quali reti (interfacce) abilitare RIP il comando è:

```
Router(config-router)#version N
```

- per abilitare/disabilitare il debug per il protocollo RIP il comando è:

```
Router#debug ip rip
Router#no debug ip rip
```

- per visualizzare la tabella di routing si utilizza:

```
Router#show ip route
```

- per visualizzare le sole entry nella tabella di routing ottenute con RIP si utilizza:

```
Router#show ip route rip
```

- per visualizzare le informazioni raccolte dal routing RIP si utilizza:

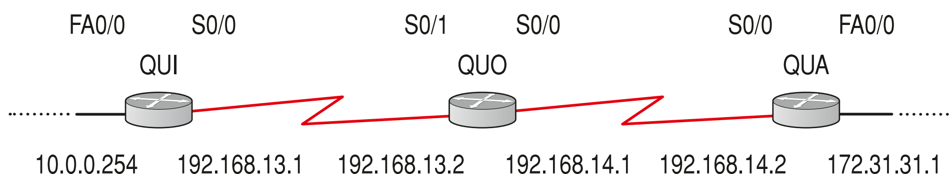
```
Router#show ip rip database
```

- per visualizzare l'elenco dei protocolli di routing attivi e il loro stato si digita:

```
Router#show ip protocols
```

ESEMPIO 20

Un esempio di configurazione con Cisco IOS per la rete di figura



è il seguente:

```
// primo router
QUI(config)#router rip
QUI(config-router)#network 10.0.0.0
QUI(config-router)#network 192.168.13.0
// secondo router
QUO(config)#router rip
QUO(config-router)#network 192.168.13.0
QUO(config-router)#network 192.168.14.0
// terzo router
QUA(config)#router rip
QUA(config-router)#network 192.168.14.0
QUA(config-router)#network 172.31.0.0
```

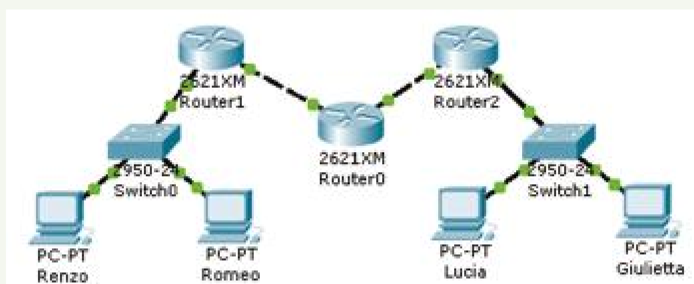
da controllare



Prova adesso!

Nella rete riportata nella seguente figura e configurata nell'esercizio di pag. 145 con i seguenti indirizzi IP assegnati alle interfacce di rete dei router e degli host:

- 192.168.1.0/24: *Renzo*, *Romeo* e *Router1 Fa0/1*
- 192.168.2.0/24: *Router1 Fa0/0* e *Router0 Fa0/0*
- 192.168.3.0/24: *Router2 Fa0/1* e *Router2 Fa0/0*
- 192.168.4.0/24: *Router1 Fa0/*, *Lucia* e *Giulietta*.



1. dopo aver eliminato tutte le rotte statiche sui router abilita il protocollo di routing RIP (versione 1) sui tre router;
2. verifica la completa connettività della rete effettuando un traceroute da Renzo e Lucia;
3. in modalità simulazione, analizza i pacchetti RIP generati dai router:
   - A. a quale indirizzo IP e porta sono destinati i pacchetti RIP?
   - B. quali reti annunciano il Router1 nei propri messaggi RIP?
   - C. viene utilizzata la procedura "Split Horizon", la procedura "Split Horizon with Poisoned Reverse" oppure nessuna delle due?

La soluzione del presente esercizio è riportata nel file reteRIP1.pkt

Esercitazioni di laboratorio 7

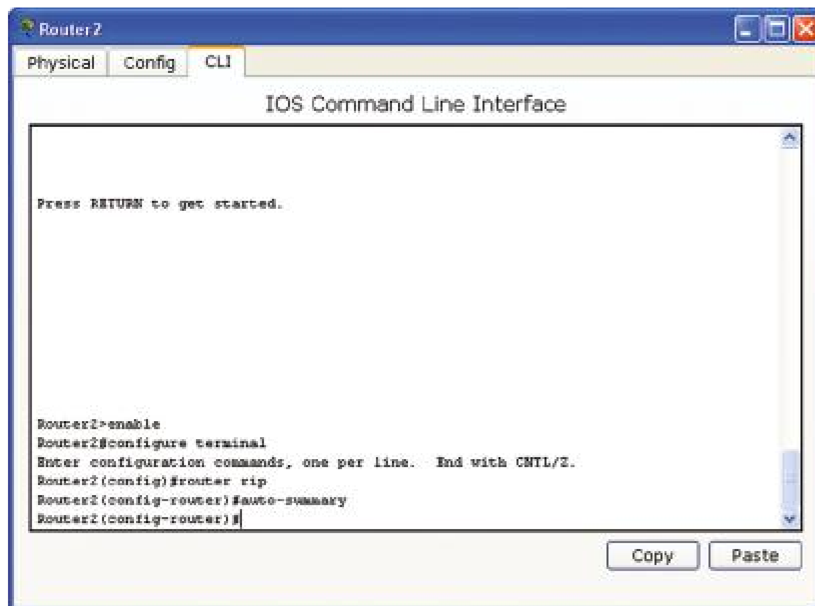
PROTOCOLLO RIPV2 E AUTO-SUMMARY

Utilizzando la funzione auto-summary vengono annunciate solo reti classful, raggruppando più sottoreti in un unica rete in modo da ridurre le dimensioni dei messaggi di routing e le dimensioni delle tabelle di routing.
Di default l'auto-summary è abilitato.

> Non si può usare l'auto-summary quando si vuole gestire il routing tra due o più sottoreti sconnesse.

Il comando che abilita l'auto-summary è il seguente:

```
Router(config-router)#auto-summary
```



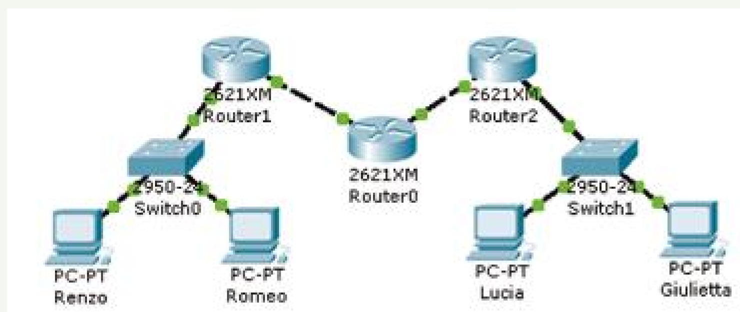
Il comando che disabilita l'auto-summary è il seguente:

```
Router(config-router)#no auto-summary
```



Prova adesso!

Nella rete riportata nella seguente figura (vedi esercitazione precedente)



A abilita sui tre router il protocollo RIPv2;

B verifica la completa connettività della rete effettuando un traceroute dal Renzo e Lucia;

C in modalità simulazione, analizza i pacchetti RIPv2 generati dai router:
- a quale indirizzo IP e porta sono destinati i pacchetti RIPv2?
- che differenze ci sono tra i messaggi RIP e quelli RIPv2?

Packet Tracer 5.x non presenta la possibilità di settare direttamente il RIPv2: è necessario inserire il comando manualmente, come in figura:

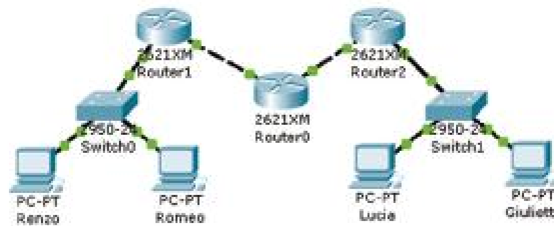
```
Router1>enable
Router1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#router rip
Router1(config-router)#version 2
Router1(config-router)#
```

La soluzione è riportata nel file reteRIP2.pkt.

Esercitazioni di laboratorio 8

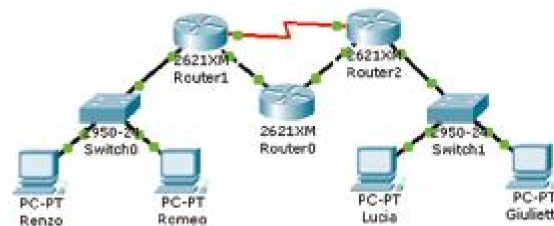
PROTOCOLLO RIPV2 E CAMBIAMENTI DI TOPOLOGIA

Si riprenda la rete riportata nella seguente figura analizzata nelle esercitazioni precedenti e presente nel file RETERIP2.pkt.



Esercizio 1

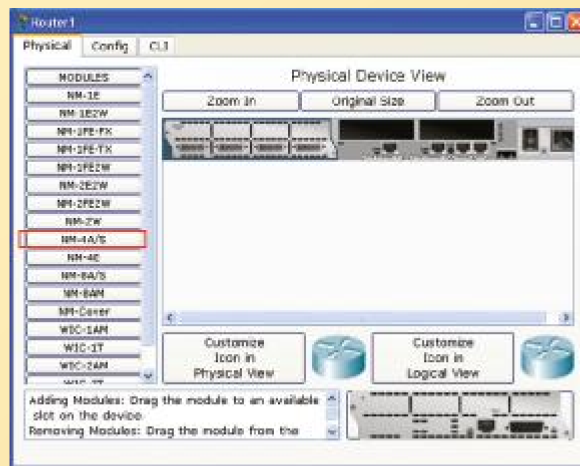
Cambia la topologia inserendo una connessione con cavo seriale, come riportata in figura:



- assegna alle interfacce di tale link indirizzi appartenenti alla rete 192.168.5.0/24;
- abilita sulle nuove interfacce il protocollo RIPv2;
- visualizza le tabelle di routing dei tre router con il comando show ip route. Quali rotte sono cambiate rispetto all'attività precedente?
- con il comando traceroute verifica quanti hop ci sono tra il Romeo e il Giulietta.
- Spegni l'interfaccia Fa0/1 del Router0 e verifica l'evoluzione delle tabelle di routing dei tre router:
  - dopo quanto tempo i tre router si accorgono del cambio di topologia? Impiegano tutti lo stesso tempo?
  - dopo quanto tempo i router rimuovono dalla propria tabella di routing la riga relativa alla rete 192.168.3.0/24?

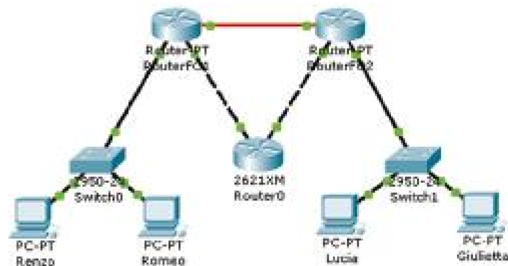
- Successivamente riattiva l'interfaccia Fa0/1 del Router0: dopo quanto tempo i tre router si accorgono del cambio di topologia?

I router Cisco 2621XM utilizzati nella simulazione di Packet Tracer non hanno la porta seriale e quindi necessitano di un modulo aggiuntivo, per esempio NM-4 A/S, come riportato in figura.
Il codice dell'esercizio è presente nel file RETERIP3.pkt.



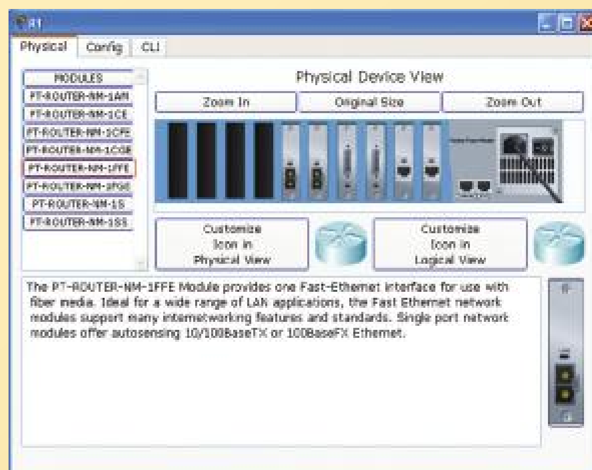
Esercizio 2

Cambia la topologia inserendo una connessione in fibra ottica, come riportata in figura:



esegui le stesse operazioni dell'esercizio precedente abilitando prima il protocollo RIPv1 e successivamente il RIPv2.

Naturalmente è necessario sostituire i router inserendone due con la porta ottica o aggiungendo un modulo, come quello della figura.
La soluzione è riportata nel file RETE-RIP3b.pkt.



3 LO STRATO DI TRASPORTO

UNITÀ DI APPRENDIMENTO

OBIETTIVI

- Capire i principi che sono alla base dei servizi del livello di trasporto:
  - multiplexing/demultiplexing
  - trasferimento dati affidabile
  - controllo di flusso e di congestione
- Descrivere i protocolli del livello di trasporto di Internet:
  - UDP: trasporto senza connessione
  - TCP: trasporto orientato alla connessione
  - controllo di congestione TCP

ATTIVITÀ

- Definire e utilizzare le porte e i socket
- Individuare gli utilizzi del protocollo UDP
- Definire il formato del segmento UDP
- Definire il formato del segmento TCP
- Utilizzare il protocollo three-way handshaking
- Stimare il valore del timeout
- Implementare i meccanismi che realizzano un trasferimento affidabile
- Individuare e risolvere i problemi connessi con l'attivazione della connessione
- Individuare e risolvere i problemi connessi con il rilascio della connessione

Lezione 1

SERVIZIO E FUNZIONI DELLO STRATO DI TRASPORTO

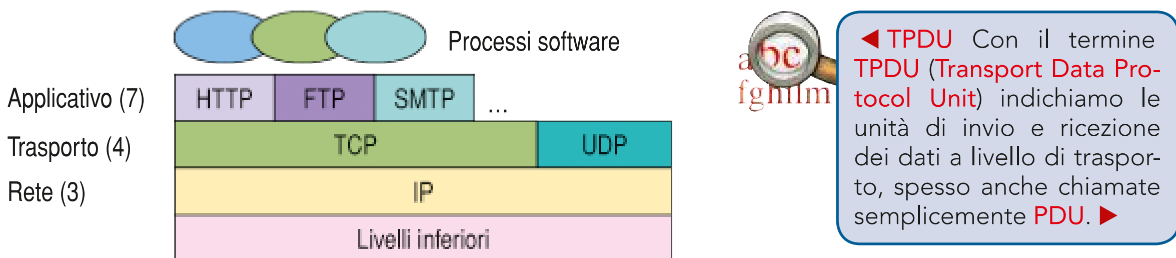
IN QUESTA LEZIONE IMPAREREMO...

- i principi che sono alla base dei servizi del livello di trasporto
- il multiplexing/demultiplexing
- l'indirizzamento di trasporto: le porte e i socket
- il concetto di trasferimento dati affidabile

Generalità

Lo strato di trasporto si colloca a livello 4 della pila ISO-OSI e svolge un compito di collegamento tra il sottostante livello di rete e i livelli applicativi, mettendo in comunicazione i diversi processi software instaurando un collegamento logico tra le applicazioni residenti su host remoti.

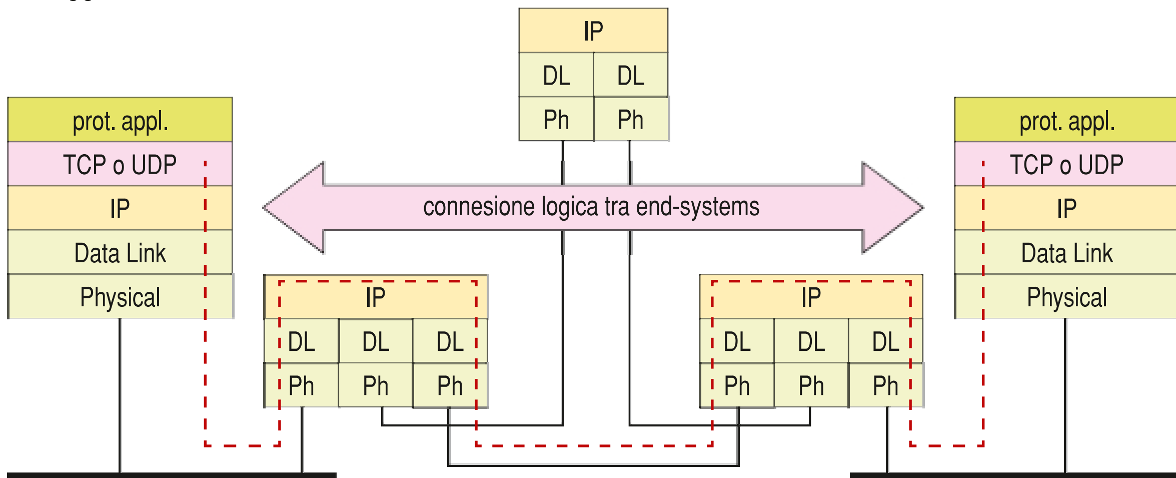
Come possiamo vedere nell'immagine seguente, nei protocolli TCP/IP i processi software (chiamati anche processi applicativi o applicazioni software) si appoggiano direttamente al livello di trasporto.



◀ TPDU Con il termine TPDU (Transport Data Protocol Unit) indichiamo le unità di invio e ricezione dei dati a livello di trasporto, spesso anche chiamate semplicemente PDU. ▶

Lo strato di trasporto, sia nel modello ISO-OSI che nel modello TCP-IP, ha il compito di mettere in connessione i livelli più bassi, orientati alla gestione del canale, con i dispositivi e i componenti hardware molto diversi tra loro, orientati al trasporto dei dati per le applicazioni dei livelli superiori. Grazie alla sua strutturazione software e ai suoi protocolli, il livello di trasporto offre le stesse primitive standard ai livelli superiori, in modo da rendere trasparente l'hardware dal software di rete. La comunicazione tra applicazione e applicazione avviene con scambio di messaggi che vengono segmentati e trasformati in ◀ TPDU ▶ (Transport Protocol Data Unit) di livello 4.

Seguiamo nella pila ISO-OSI il percorso delle PDU inviate da una applicazione: il livello di trasporto passa le PDU al livello di rete che le incapsula in PDU di livello 3 e le inoltra; dopo alcuni passaggi intermedi giungono al livello 3 del destinatario che li passa al livello 4 dove vengono ricostruiti dell'applicazione destinazione.



Il livello di trasporto rende trasparente il trasporto fisico dei messaggi alle applicazioni.

ESEMPIO 1

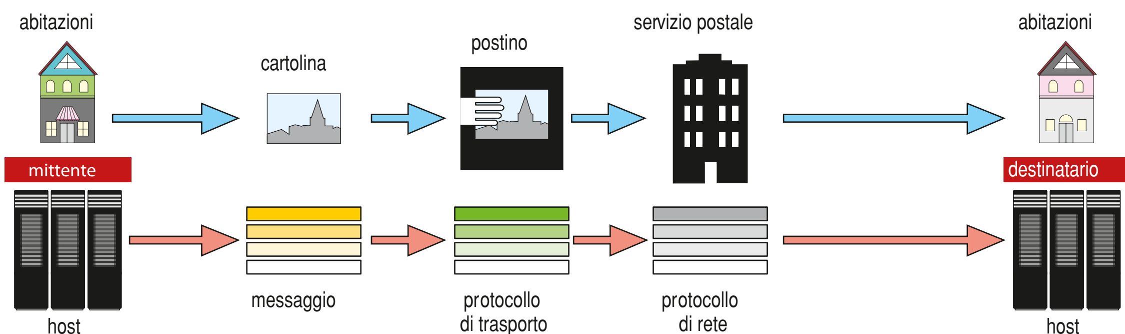
L'esempio dell'invio di una lettera attraverso il servizio postale fornisce una valida analogia. Supponiamo di voler spedire una cartolina: possiamo individuare i seguenti componenti di sistema e i relativi elementi di rete:

- mittente/destinatario : processi (applicazioni livelli superiori)
- abitazioni : host
- cartolina : messaggi (TPDU)

La cartolina viene imbucata nella cassetta delle lettere da cui verrà ritirata dal servizio postale per essere portata nella centrale di smistamento. Dopo alcuni passaggi tra sportelli e filiali, la cartolina verrà consegnata all'ufficio postale della città dove risiede il destinatario per essere poi collocata nella casella postale del destinatario.

Individuiamo in questo sistema:

- postini : protocollo di trasporto
- servizio postale : protocollo del livello di rete



I servizi del livello di trasporto

Innanzi tutto dobbiamo individuare la differenza tra servizio e protocollo:

- servizio: è l'insieme delle operazioni primitive che un livello mette a disposizione al livello superiore;
- protocollo: è l'insieme delle regole che governano il formato e il significato delle informazioni che le ◀ entity ▶ si scambiano fra loro. Le entity usano i protocolli per implementare i propri servizi.

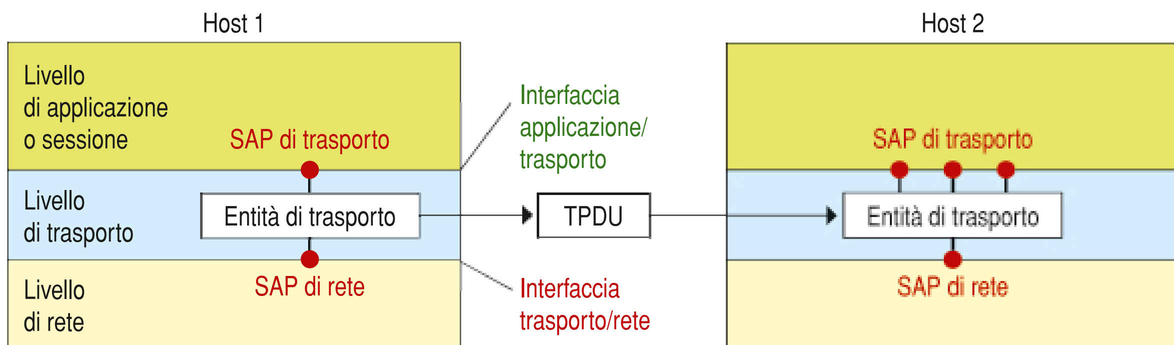


◀ Entity Si definisce entità (entity) di rete un elemento o processo che effettua una conversazione in rete. Le entità (processi) che effettuano una conversazione con un elemento di pari livello si chiamano peer entitiy (entità di pari livello). ▶

Con ◀ **Service Access Points** ▶ si indentifica l'interfaccia logica tra una entity di livello (N-1) e una entity di livello (N) nella pila ISO-OSI. In altre parole, un Service Access Point è il punto di accesso a un servizio che un livello offre al suo soprastante.



◀ **SAP** A Service Access Point (SAP) is an identifying label for network endpoints used in Open Systems Interconnection (OSI) networking. The SAP is a conceptual location at which one OSI layer can request the services of another OSI layer. ▶



I protocolli dello strato di trasporto effettuano le seguenti funzioni di base:

- il protocollo di trasporto residente nel terminale sorgente (host1) riceve messaggi dall'applicazione mittente mediante il SAP di trasporto, li incapsula in segmenti e invia la sequenza ordinata di segmenti TPU nel canale logico: questa operazione è detta multiplazione di segmenti;
- il protocollo di trasporto residente nel terminale di destinazione (host2), riceve la sequenza di segmenti in uscita dal canale logico mediante il SAP di trasporto, frammenta la sequenza in segmenti, estrae da ciascun segmento il corrispondente messaggio, passa il messaggio al processo destinazione residente nel terminale host2: questa operazione è detta demultiplazione di segmenti.

I servizi offerti dallo strato di trasporto sono sostanzialmente di due tipi:

- trasporto affidabile: garantisce la consegna dei messaggi nel corretto ordine;
- trasporto non affidabile: garantisce solo funzionalità di indirizzamento.



SERVIZIO AFFIDABILE E NON AFFIDABILE

Un servizio è affidabile quando non perde mai i dati, cioè quando:

- tutti i dati spediti arriveranno al destinatario, oppure viene segnalato un errore;
- il ricevente invia un acknowledgment (ACK di conferma) al mittente per ogni pacchetto ricevuto.

Un servizio si dice invece non affidabile se non offre alcuna certezza che i dati inviati siano effettivamente ricevuti.

Lo strato di trasporto della rete Internet mette a disposizione delle applicazioni attive in ciascun host due distinti protocolli di trasporto:
- TCP (Transmission Control Protocol);
- UDP (User Datagram Protocol).

TCP e UDP svolgono funzioni diverse, cioè offrono servizi diversi allo strato applicativo: ogni applicazione (processo) in esecuzione, prima di poter trasmettere messaggi sulla rete, sceglie il protocollo da usare per poter realizzare la comunicazione. La differenza fondamentale tra i due protocolli, che saranno descritti dettagliatamente nelle prossime due lezioni, è che il protocollo TCP è orientato alla connessione (connection-oriented), ed è quindi affidabile in quanto consente il controllo dell'integrità dell'informazione contenuta nei pacchetti, inoltre possiede un sistema per la segnalazione dell'errore al mittente (scambio di messaggi di ◀ **acknowledge** ▶), mentre il protocollo UDP è senza connessione (connectionless) e quindi non affidabile.



◀ **Acknowledge** An acknowledge character (ACK) is a transmission control character transmitted by the receiving station as an acknowledgement, an affirmative response to the sending station. ▶



Zoom su...

CONNECTION-ORIENTED E CONNECTIONLESS

Nella comunicazione connection-oriented prima di poter iniziare a effettuare la trasmissione è necessario provvedere alla connessione tra mittente e destinatario e individuare anche gli intermediari che permettono di realizzarla (router intermedi): la comunicazione può avvenire in modo bidirezionale e offre una elevata qualità e maggiori garanzie, naturalmente con costi più elevati.
La comunicazione connectionless è adatta per dati occasionali, quando non è richiesta una elevata qualità della comunicazione e non è neppure necessario il rispetto della sequenza nella ricezione dei messaggi, dato che in questa modalità un messaggio trasmesso successivamente può arrivare prima di uno precedente.
Il costo di questa modalità è limitato ma sono scarse le garanzie che offre.

Primitive a livello di trasporto

I protocolli di trasporto sono implementati nei più diffusi sistemi operativi e forniscono ai programmatori le funzioni di base (◀ primitive ▶) per poter usare i protocolli stessi e far comunicare processi remoti. Un servizio di livello *n* è definito formalmente dalle primitive che una entità di livello *n* + 1 può adoperare per accedere al servizio stesso.

Due applicazioni cooperano tramite le primitive messe a disposizione dal livello inferiore per implementare le funzionalità del livello cui appartengono; le primitive base fornite dal livello 4 alle applicazioni sono:
- LISTEN: si mette in attesa di una richiesta di connessione
- SEND DATA: per trasmettere un contenuto
- RECEIVE DATA: per ricevere un contenuto
- T-CONNECT: per aprire la connessione
- T-DISCONNECT: per chiudere la connessione



◀ Primitive Con il termine primitive si intendono sia le funzioni di base messe a disposizione del programmatore dal linguaggio di programmazione che le funzioni del kernel richiamabili dal sistema operativo. ▶

e vengono utilizzate dal trasmettitore con la seguente tipica sequenza:
- T-CONNECT: attiva la connessione con negoziazione
- n SEND DATA: invio di n segmenti di dati
- T-DISCONNECT: chiusura della connessione

Per ogni primitiva sono generalmente disponibili i seguenti metodi:

Metodo primitiva	Descrizione
request()	si chiede al servizio di fare qualcosa (una azione)
indication()	si viene avvertiti dal servizio di un evento (segnalazione di evento)
response()	si chiede al servizio di rispondere a un evento (una azione)
confirm()	il servizio segnala l'arrivo di una conferma (segnalazione di evento)

La sintassi per l'utilizzo delle primitive è la seguente:

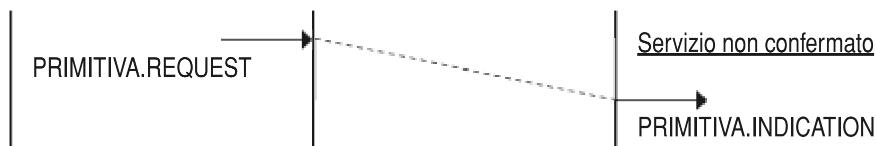
```
<Nome primitiva> <punto> <Tipo primitiva>
```

Per esempio:

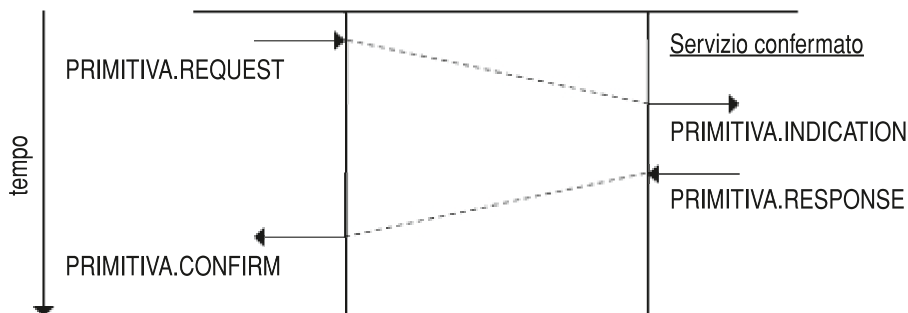
```
T-CONNECT.response()
```

Nel dialogo tra due host si possono utilizzare le forme:

A connessione asincrona: nessuna conferma di ricezione al mittente (connection-less);



B connessione sincrona: con conferma e azione al mittente (connection-oriented).

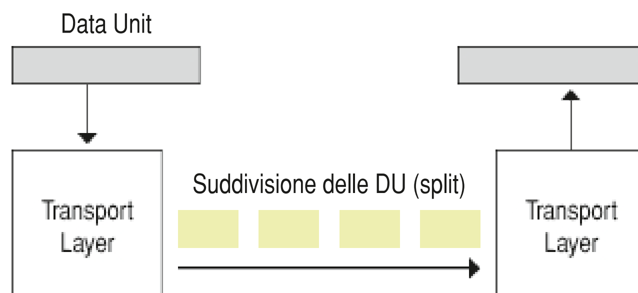


Per esempio, per stabilire una connessione sincrona fra A e B si eseguono le seguenti opzioni:

Entità	Azione	Descrizione
A	invia T-CONNECT.request()	A vuole connettersi
B	riceve T-CONNECT.indication()	qualcuno vuole connettersi a B
B	invia T-CONNECT.response()	B accetta la connessione
A	riceve T-CONNECT.confirm()	Conferma di connessione stabilita

Il multiplexing/demultiplexing

Quando più applicazioni sono attive, sia su diversi end-system che su diversi host, circolano in rete più messaggi generati dalle diverse applicazioni: il livello di trasporto genera dei canali logici tra le diverse applicazioni svolgendo la funzione di ◀ multiplexing/demultiplexing ▶.





◀ Multiplexing/demultipexing Con il termine multiplexing (o multiplazione) si intende la raccolta dei dati provenienti dai processi di applicazione e il loro imbustamento con intestazione (header), necessaria per la successiva operazione opposta di demultiplexing e il relativo passaggio della sequenza ordinata dei segmenti allo strato di rete. Con il termine demultiplexing (o demultiplazione) si intende la consegna dei segmenti ricevuti a opportuni processi di livello applicazione: la sequenza di dati ricevuta dallo strato di rete viene frammentata in segmenti, viene individuato l'indirizzo di destinazione presente nell'header di ciascun segmento e viene effettuata la loro consegna. ▶

Per effettuare il multiplexing/demultiplexing non è però sufficiente aggiungere ai segmenti l'indirizzo IP in quanto è necessario aggiungere un identificativo che ci permette di ricostruire il dato tra tutti quelli che possono arrivare al medesimo ricevente a opera dello stesso mittente da parte di applicazioni diverse: è necessario introdurre un nuovo tipo di indirizzamento, che prende il nome di indirizzamento di trasporto.

L'indirizzamento di trasporto

Per poter risolvere il problema della trasmissione dei dati tra applicazioni diverse sui medesimi host, il protocollo di trasporto utilizza il meccanismo delle ◀ porte ▶.



◀ Porte Per i protocolli a livello di trasporto, viene introdotto il concetto di porta. Una porta (port) è un valore numerico specificato su 2 byte (da 0 a 65535) che identifica un particolare canale utilizzabile per la comunicazione. In questo modo è possibile instaurare simultaneamente più comunicazioni, cosicché due applicazioni possono comunicare l'una con l'altra indipendente dal fatto che sulla rete stiano avvenendo altre comunicazioni: è sufficiente utilizzare porte diverse. Si viene così a delineare il socket che rappresenta un punto di connessione per una comunicazione, identificato univocamente da un indirizzo IP e un numero di porta, ovvero da una coppia indirizzo IP: porta. Un'applicazione che vuole comunicare con un'altra utilizzando un protocollo a livello di trasporto, deve creare un socket locale formato da:

indirizzo IP locale:porta locale,

comunicando al sistema operativo di voler utilizzare una determinata porta, in modo tale che tutti i pacchetti a essa destinati le vengano recapitati. L'applicazione invierà i pacchetti a un determinato socket di destinazione formato da:

indirizzo IP di destinazione:porta di destinazione.

I pacchetti verranno ricevuti dall'interfaccia di rete il cui indirizzo IP è quello specificato come destinazione all'interno dei pacchetti stessi e il sistema ricevente recapiterà i pacchetti ricevuti all'applicazione a cui si riferisce la porta di destinazione. ▶

È bene sottolineare il fatto che i numeri di porta sono relativi soltanto al protocollo considerato: una determinata porta per il protocollo TCP è diversa dallo stesso numero di porta per il protocollo UDP (si tratta effettivamente di porte diverse), anche se in genere viene utilizzato lo stesso numero di porta per un servizio che gestisce entrambi i protocolli.

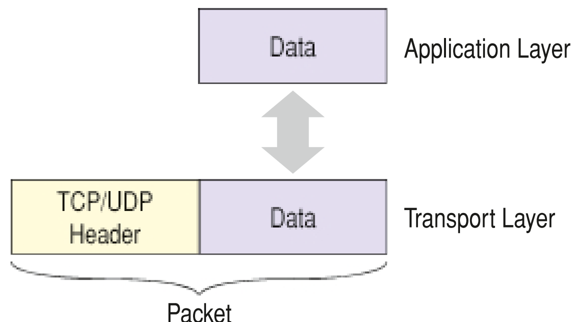
ESEMPIO 2

Riprendendo l'analogia con il servizio postale, supponiamo di spedire una lettera a un amico che vive in condominio: non è sufficiente mettere l'indirizzo di destinazione ma è necessario indicare il cognome e, se non vive da solo, anche il suo nome. Per individuare univocamente il destinatario è necessario specificare quindi oltre al suo indirizzo anche il suo nome completo: gli stessi dati sono a lui necessari affinché possa darci una risposta e quindi li indicheremo sul retro della busta, come indirizzo del mittente.

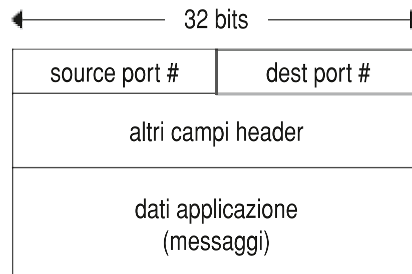
A ogni macchina viene associato un insieme di porte, i flussi di dati distinti all'interno della stessa macchina sono caratterizzati da porte diverse, cioè vengono associati ciascuno a una specifica porta.

Una connessione tra due computer viene quindi univocamente identificata dalle coppie:
1 "indirizzo IP: porta" del mittente
2 "indirizzo IP: porta" del destinatario

I valori delle porte sono scritti nell'header che viene aggiunta ai dati dallo strato di trasporto.



A livello 4 il segmento TCP/UDP è costituito da una intestazione (header) e da un campo dati (payload) con la struttura a fianco: ▶



Nell'header, oltre alla porta del mittente (search port number) e alla porta del destinatario (target port number), vengono scritte delle informazioni di controllo come una sequenza di numeri di controllo e una cecksum per il controllo d'integrità del dato.

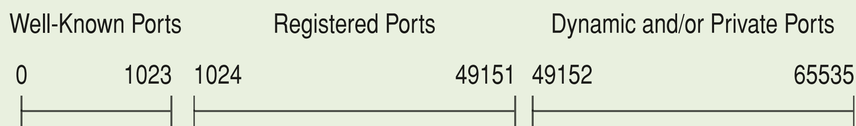
Come vedremo in seguito, gli header dell'UDP hanno una dimensione di 8 bytes mentre quelli del TCP dai 20 ai 24 bytes e la dimensione massima (in byte) del campo dati viene fissata, di volta in volta, al momento in cui viene stabilito il canale logico tra i due processi comunicanti. La dimensione massima è detta Maximum Segment Size (MSS) ed è espressa in byte.



Zoom su...

NUMERI DI PORTA TCP/UDP

I numeri di porta da 0 a 1023 sono riservati ad applicazioni particolari (quali HTTP, FTP, DNS, TelNet ecc.), costituiscono i cosiddetti "well-known port numbers" e possono essere usati solo dai server in apertura passiva. I numeri di porta da 1024 a 49151 sono riservati a porte registrate e sono usati da alcuni servizi ma possono anche essere utilizzate dai client (tutti i client usano normalmente le porte a partire dalla numero 1024 per collegarsi a un sistema remoto).
I numeri di porta da 49152 a 65535 sono liberi per essere assegnati dinamicamente dai processi applicativi.



La lista completa dei "well-known port numbers" è reperibile al sito http://www.iana.org/assi-gnments/port-numbers. Di seguito sono riportati alcuni esempi.

Porta logica	Protocollo di rete
7	ECHO
21/tcp	FTP (file transfer protocol)
22/tcp	SSH (Secure SHell)
23/tcp	TELNET
25/tcp	SMTP (Simple Mail Transfer Protocol)
42	WINS (Windows Internet Naming Service)
53	DNS (Domain Name Service)
80/tcp	HTTP (Hyper Textual Transfer Protocol)
110/tcp	POP3 (Post Office Protocol, v3)
143/tcp	IMAP (Internet Message Access Protocol)
161	SNMP (Simple Network Management Protocol)
389/tcp	LDAP (Lightweight Directory Access Protocol)
443/tcp	HTTPS (Secure HTTP)

ESEMPIO 3

Nell'esempio di figura, due applicazioni dell' host A

host A: <137.204.10.85:3300>
host A: <137.204.10.85:3301>

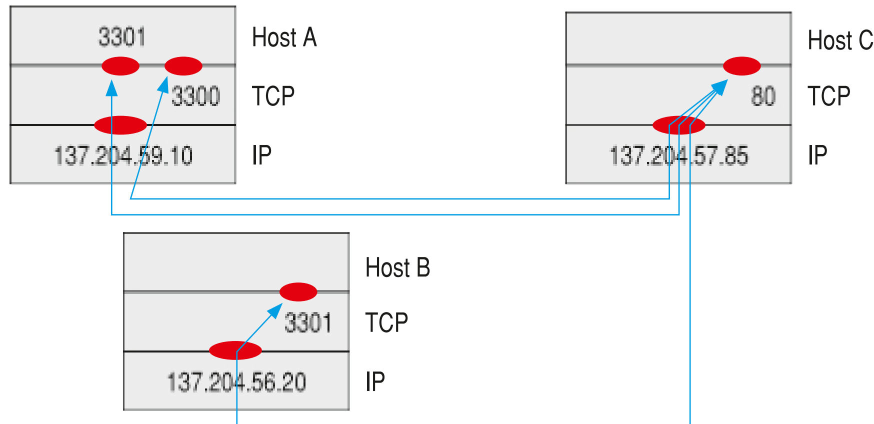
e una dell'host B

host B: <137.204.56.10:3301>

si connettono alla stessa porta 80 dell'host C richiedendo un servizio HTTP:

host C: <137.204.57.85:80>

non possono essere distinti solo dal loro indirizzo IP ma sono univocamente individuati mediante l'indirizzo del socket.



I processi client/server

Come abbiamo visto sugli host possono essere eseguite applicazioni che scambiano messaggi con altre applicazioni remote: affinché un calcolatore possa effettivamente utilizzare un messaggio in arrivo, è necessario che in quel momento vi sia in esecuzione un processo in attesa di quel messaggio e che lo possa interpretare. È il caso tipico di applicazioni internet, dove la maggior parte dei servizi telematici offerti si basano su questa modalità di architettura che prende il nome di ◀ **client-server** ▶.



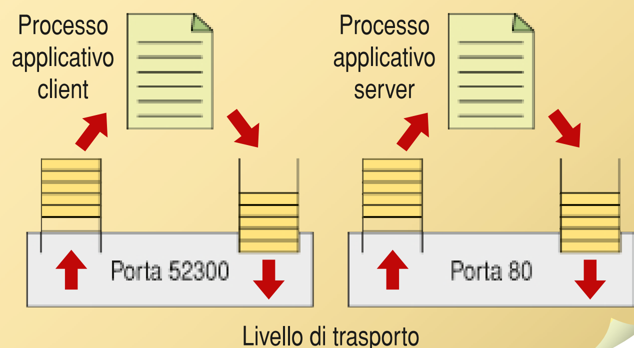
◀ **Client/server** The client/server model is a computing model that acts as a distributed application wich partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. ▶

Il modello client-server è organizzato in due moduli, chiamati appunto server e client, operanti su macchine diverse:

- il server svolge le operazioni necessarie per realizzare un servizio;
- il client, generalmente tramite una interfaccia utente, acquisisce i dati, li elabora e li invia al server richiedendo un servizio.

Quindi in rete sono presenti calcolatori su cui girano processi Server che erogano servizi e sono in attesa di ricevere richieste di connessione da parte di processi Client interessati a usufruire di tali servizi.

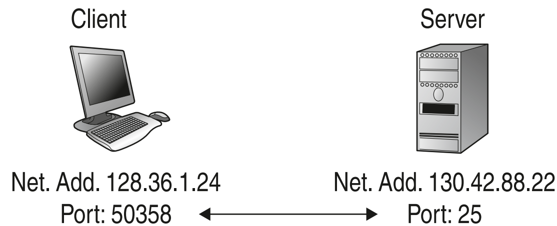
Il client deve conoscere l'indirizzo IP e il numero di porta usati dal server per potersi collegare al socket di destinazione: per esempio, il servizio HTML del server è sulla porta 80 mentre quello del client è libero, a scelta tra quelli disponibili (per esempio 52300), ma che deve essere trasmesso al server nel segmento di richiesta in modo che il server possa successivamente inviare la risposta.



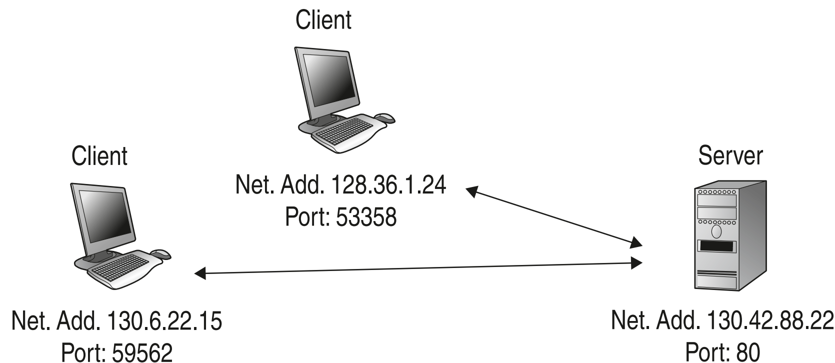
Naturalmente allo stesso server arrivano contemporaneamente richieste da client diversi che potrebbero anche utilizzare la stessa porta, come nel seguente esempio la 3301, o richieste da porte diverse dello stesso host: il collegamento avviene univocamente tramite i socket.

ESEMPIO 4

Un client si connette alla porta di un server SMTP remoto (servizio di posta elettronica)



Due client accedono alla stessa porta di un server HTTP; non c'è comunque ambiguità, perché la coppia di socket è diversa



> Quando un client apre un socket, non dovrebbe utilizzare una porta nota, in quanto riservata per l'offerta di servizi.

ESEMPIO 5

Riportiamo di seguito due frammenti di codice tipici di un'applicazione client-server:

Server	Client
	`...`
`...`	`connect();`
`accept();`	`receive();`
`send();`	`send();`
`receive();`	`...`
`...`	`disconnect();`

La chiamata di queste primitive in questa sequenza permette di realizzare una comunicazione senza errori, come vedremo nelle prossime lezioni.

Qualità del servizio QoS

Il livello di trasporto ha anche la funzione di aumentare la qualità del servizio offerto dalla rete e introduce il QoS (Quality of Service), che descrive il livello delle prestazioni di rete che deve essere assicurato per una particolare applicazione.
L'utilità di valutare la qualità del servizio è necessaria soprattutto sulle reti di medie/grandi dimensioni dove è presente un notevole flusso di dati e le applicazioni necessitano di tempi di risposta diversa (alcune richiedono qualità elevata).

ESEMPIO 6

Se un server di posta consegna una mail con un minuto di ritardo nessuno se ne accorge e questo ritardo non crea nessun problema oggettivo: per questa trasmissione basterà un basso livello di QoS. Se invece una applicazione di video streaming trasmette ogni frame video con un ritardo di mezzo secondo il filmato si visualizza a scatti e risulta quindi inaccettabile: per questa trasmissione è necessario il più alto livello di QoS.

È possibile specificare il QoS come un range di valori entro il quale definire la qualità del servizio per differenziare a dovere le diverse necessità, e può essere definito in modo:
- assoluto: vengono stabiliti i valori che devono essere rispettati da un insieme di parametri indicatori delle prestazioni (per esempio il ritardo massimo, la probabilità di perdita di pacchetti ecc.);
- relativo: si definiscono le modalità di trattamento di una classe di traffico rispetto alle altre (per esempio il livello di priorità di servizio, il livello di priorità di scarto ecc.).

Riportiamo di seguito l'elenco dei parametri indicatori che permettono di dare la definizione standard del QoS:
- ritardo massimo nell'attivazione della connessione;
- numero di byte trasferiti nell'unità di tempo (throughput);
- velocità di consegna o ritardo di transito;
- probabilità di fallimento della connessione o di interruzione in caso di congestione;
- probabilità che la connessione non venga stabilita entro il massimo tempo di ritardo di attivazione;
- tasso di errore residuo o numero di messaggi persi sul numero totale di messaggi inviati;
- probabilità di fallimento del trasferimento;
- ritardo di rilascio della connessione e probabilità di fallimento nel rilascio della connessione;
- protezione contro le intercettazioni dati (lettura o modifica non autorizzata);
- priorità della connessione;
- probabilità che il livello di trasporto termini la connessione per problemi interni o di congestione.

Prima di effettuare una connessione tra due macchine avviene quella che prende il nome di negoziazione delle opzioni: le stazioni trasmissive negoziano all'inizio per concordare i parametri da utilizzare e se non raggiungono un accordo la connessione può anche essere rifiutata.

La negoziazione segue questo schema:

A. l'utente quando vuole stabilire una connessione comunica al livello di trasporto i parametri di QoS a lui necessari;
B. il livello di trasporto esamina i parametri di qualità richiesti dall'utente e
   - se li può garantire attiva la connessione;
   - se si rende conto di non essere in grado di soddisfarli
     – non inizia la connessione e comunica all'utente il fallimento della stessa;

– richiede alla macchina di destinazione di stabilire una connessione e le invia una contro-proposta indicando i valori che è in grado di offrire: se la proposta è accettata dall'utente, la connessione può essere attivata; in alternativa è l'utente a fare una controrichiesta a livelli più bassi e così via.

È importante sottolineare che i parametri QoS negoziati alla fine della trattativa restano validi e fissati per tutta la durata della connessione.

Servizi offribili dallo strato di trasporto

Ricapitoliamo i principali servizi che possono offrire i protocolli di uno strato di trasporto prima di analizzare in dettaglio i due protocolli di riferimento, il TCP e l'UDP:

- multiplazione/demultiplazione: lo strato di trasporto riceve i messaggi generati da uno o più processi attivi nel terminale sorgente e li ordina in una sequenza di segmenti (multiplazione): a destinazione riceve la sequenza di segmenti e ne estrae le sotto-sequenze di messaggi da trasferire a una o più applicazioni attive nel terminale di destinazione (demultiplazione);
- gestione della connessione: offre il servizio di instaurazione, gestione e chiusura della connessione tra due processi comunicanti attraverso un canale logico;
- rivelazione di errori: il mittente immette in ciascun segmento da trasmettere una sequenza di bit di controllo della parità che vengono utilizzati in ricezione per rivelare la presenza di eventuali errori;
- trasferimento affidabile di segmenti: attua le tecniche necessarie a rendere affidabile il canale logico che viene creato tra due processi, facendo in modo di eliminare gli errori, le perdite e le duplicazioni dei messaggi scambiati;
- controllo del flusso dei segmenti trasmessi: gestisce il flusso dei segmenti inviati dal processo mittente in modo da evitare che il buffer presente allo strato di trasporto vada in overflow;
- controllo di congestione: analogamente al precedente, ha lo scopo di prevenire l'overflow dei buffer presenti nei router e nei bridge della rete dovuto agli scambi di segmenti tra tutte le coppie di processi comunicanti sulle rete regolando la velocità con cui i segmenti stessi vengono immessi nella rete.

La differenza fondamentale tra UDP e TCP consiste nel fatto che:

- UDP fornisce un servizio di trasferimento di segmenti privo di connessione e non affidabile;
- TCP fornisce un servizio di trasferimento dei segmenti orientato alla connessione e affidabile.

In particolare:

UDP fornisce solo i due seguenti servizi:	TCP fornisce i seguenti servizi:
◗ multiplazione/demultiplazione; ◗ rivelazione degli errori.	◗ multiplazione/demultiplazione; ◗ rivelazione degli errori; ◗ trasferimento affidabile; ◗ controllo di flusso; ◗ controllo di congestione; ◗ gestione della connessione.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Con TPDU si intende:

a) Transport Data Protocol User
b) Transport Device Protocol User
c) Transport Device Protocol Unit
d) Transport Data Protocol Unit

2 Con SAP si indentifica:

a) l'interfaccia logica tra entity di livello diverso
b) l'interfaccia logica tra entity di livello uguale
c) il punto di accesso a Internet
d) il punto di accesso ai servizi di rete

3 Quale dei seguenti acronimi è errato?

a) TCP Transmission Control Protocol
b) UDP User Datagram Protocol
c) SAP Server Access Point
d) OSI Open Systems Interconnection

4 Col metodo indication():

a) si chiede al servizio di fare qualcosa (una azione)
b) si viene avvertiti dal servizio di un evento
c) si chiede al servizio di rispondere a un evento
d) il servizio segnala l'arrivo di una conferma

5 Il multiplexing esegue le seguenti operazioni:

a) effettua raccolta dati dai processi di applicazione
b) indirizza i dati sulla porta specificata
c) effettua il loro imbustamento con header
d) passa la sequenza ordinata dei segmenti allo strato di rete

6 Il Maximum Segment Size indica:

a) la dimensione massima del campo dati
b) la dimensione massima del campo header
c) la dimensione massima del segmento
d) la dimensione massima del socket

7 Associa le seguenti porte (7,21,22,80,110) ai servizi:

a) .... HTTP
b) .... FTP
c) POP3
d) .... ECHO
e) .... TELNET

8 Indica quali dei seguenti servizi sono forniti dall'UDP, quali dal TCP e quali da entrambi.

Servizio	UDP	TCP
multiplazione/demultiplazione	○	○
trasferimento affidabile	○	○
controllo di flusso	○	○
rivelazione degli errori	○	○
controllo di congestione	○	○
gestione della connessione	○	○

Test vero/falso

1 Il livello di trasporto rende trasparente il trasporto fisico dei messaggi alle applicazioni. V F
2 Le entity usano i protocolli per implementare i propri servizi. V F
3 Il protocollo TCP è senza connessione (connectionless). V F
4 Il protocollo TCP possiede un sistema per la segnalazione dell'errore al mittente. V F
5 La connection-oriented non richiede il rispetto della sequenza nella ricezione dei messaggi. V F
6 Nella connessione sincrona avviene la conferma dell'azione al mittente. V F
7 Le porte logiche permettono di gestire flussi multipli di dati su una sola connessione fisica. V F
8 Un socket è composto da un numero di porta di un processo e dall'indirizzo IP del terminale. V F
9 Nell'hedaer il numero di porta dell'applicazione sorgente che ha generato il messaggio è di 16 bit. V F
10 Prende il nome di indirizzo del socket un numero di porta concatenato a un indirizzo IP. V F

Esercizi di completamento

1 Il livello di trasporto rende trasparente il .... dei messaggi alle applicazioni.

2 Le entità che effettuano una conversazione con un elemento di pari livello si chiamano ....

3 Un servizio è .... quando non perde mai i dati, cioè quando:

a) tutti i dati spediti arrivano al ...., oppure viene ....;

b) il ricevente invia un .... al mittente per ogni pacchetto ricevuto.

4 Un servizio si dice invece .... se non offre alcuna certezza che i dati inviati siano effettivamente ricevuti.

5 Il socket rappresenta un punto di connessione per una comunicazione, identificato univocamente da .... e un ....

6 Un'applicazione che vuole comunicare con un'altra utilizzando un protocollo a livello di trasporto, deve creare un .... locale formato da ....

7 Nell'header TCP/UDP, oltre alla .... e alla .... vengono scritte delle informazioni di controllo.

8 Il modello client-server è organizzato in due moduli, chiamati appunto server e client, operanti ....:

a) il server svolge le operazioni necessarie ....;

b) il client, generalmente tramite una interfaccia utente, ... e li invia al server ....

Lezione 2

IL PROTOCOLLO UDP

IN QUESTA LEZIONE IMPAREREMO...

- le caratteristiche del protocollo UDP
- gli utilizzi del protocollo UDP
- il formato del segmento UDP

Generalità

Il protocollo UDP (User Datagram Protocol) è stato concepito per tutte quelle applicazioni per le quali non è necessaria una completa gestione delle connessioni, come per esempio le trasmissioni televisive in streaming dove può essere tranquillamente perso qualche fotogramma senza pregiudicare la visione del programma. Tale protocollo utilizza il protocollo IP per trasportare i messaggi ma, rispetto a quest'ultimo, distingue tra più destinazioni all'interno di uno stesso host mediante il meccanismo delle porte (socket). Viene utilizzato pertanto da tutte le applicazioni che trasmettono pacchetti singoli, senza necessità di ◀ **acknowledgment** ▶ né di ◀ handshaking ▶ tra mittente e destinatario.



◀ Handshaking Indica la fase preliminare con la quale i dispositivi definiscono i protocolli e le velocità da utilizzare per poter comunicare tra loro ed è composta da uno scambio di segnali prestabiliti tra i due dispositivi stessi: l'handshake è necessario tra protocolli connection-oriented mentre non è necessario nei protocolli connectionless. ▶

◀ **Acknowledgment** Message such as one used in 'handshaking' process between two host that indicates the status of communications received: commonly written as ACK. ▶



Il protocollo UDP è inoltre adatto a essere impiegato dalle applicazioni che richiedono bassi ritardi di trasferimento end-to-end ma tollerano perdite/errori nei segmenti ricevuti dai processi di destinazione. La tabella seguente riporta le più comuni applicazioni e i relativi protocolli utilizzati nello strato di applicazione, che sarà descritto nell'unità di apprendimento 4.

Applicazione	Protocollo di strato applicativo
Telefonia via Internet	Voip
Applicazioni Multimediali	—
Protocolli di instradamento	RIP
Risoluzione di nomi	DNS
Amministrazione di rete	SNMP
File server remoti	NFS
Trivial File Transfer	tftp
Network Time Protocol	ntp

Il segmento UDP

Ogni segmento UDP fa uso di datagrammi con un'intestazione di 8 byte, è gestito indipendentemente dagli altri e può dare luogo a una modalità di trasferimento tra processi di tipo:

- punto-punto;
- bidirezionale;
- full-duplex.

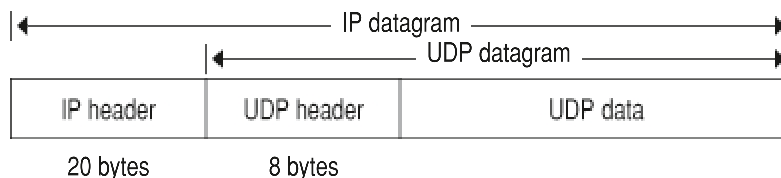
A differenza del protocollo TCP, il protocollo UDP è in grado di supportare il ◀ multicast ▶, cioè la distribuzione simultanea di informazione verso un gruppo di destinatari.



◀ Multicast Multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source. Copies are automatically created in other network elements, such as routers, but only when the topology of the network requires it. ▶

Ogni datagramma UDP viene incapsulato in un datagramma IP, quindi la dimensione del datagramma UDP non può superare la dimensione massima della parte dati del datagramma IP ed è formato da due parti:

- un header (8 byte);
- un campo ◀ payload ▶, di lunghezza variabile ma non superiore al ◀ **Maximum Segment Size (MSS)** ▶.



Il valore di default Maximum Segment Size è di 536 byte mentre il valore massimo è 65535 byte: la dimensione è decisa dal mittente durante la fase di setup della comunicazione.

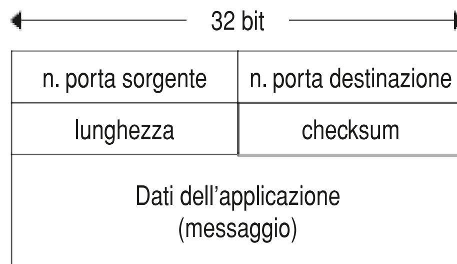


◀ Payload Indica il carico utile che viene ricevuto dal sistema destinazione, cioè i dati contenenti le informazioni al netto di tutti i campi legati al protocollo e necessari affinché avvenga la comunicazione. ▶



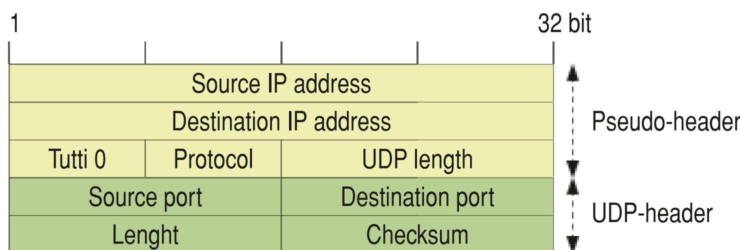
◀ **Maximum Segment Size (MSS)** The Maximum Segment Size (MSS) is the largest amount of data, specified in bytes, that a computer or communications device can handle in a single, unfragmented piece. For optimum communications, the number of bytes in the data segment and the header must add up to less than the number of bytes in the Maximum Transmission Unit (MTU). ▶

Il formato del segmento UDP è il seguente:



Si può osservare che non sono presenti gli indirizzi IP in quanto questi sono già presenti nel header IP che lo incapsula.

Il campo header è composto da 4 elementi:
- Source/Destination Port: numero di porta sorgente e destinazione, utilizzati nelle operazioni di multiplazione/demultiplazione;
- Length: lunghezza in byte del segmento UDP, comprendente anche i byte di intestazione;
- Checksum: controllo degli errori su intestazione e dati, che viene calcolato ponendo in testa la cosiddetta pseudo intestazione UDP così strutturata:



Il checksum viene calcolato coinvolgendo anche gli indirizzi IP, per verificare che il datagramma UDP sia effettivamente arrivato al giusto indirizzo IP di destinazione.

La pseudo intestazione non viene trasmessa, serve unicamente per il calcolo del checksum.

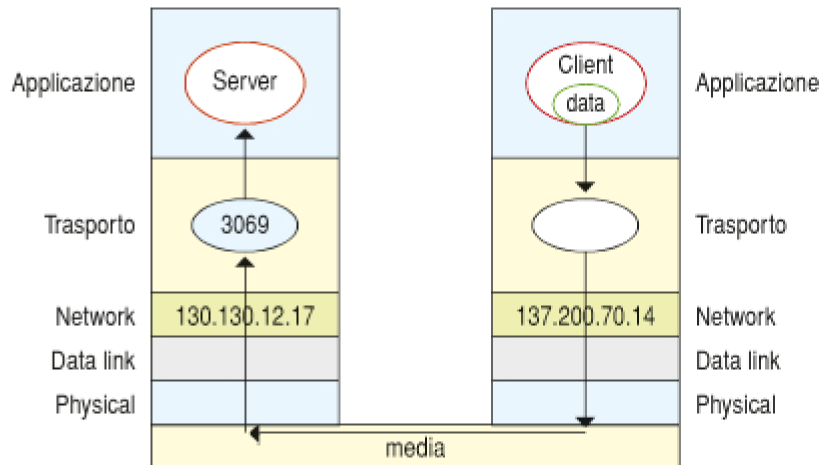
La multiplazione/demultiplazione in UDP

Il servizio di trasferimento di messaggi privo di connessione e le operazioni di multiplazione/demultiplazione effettuate attraverso il canale logico tra un processo sorgente e un processo destinazione vengono realizzate mediante la coppia di parametri:

<indirizzo IP destinatario: Porta del destinatario>

Descriviamo un semplice esempio di una applicazione che sfrutta i socket per trasmettere un datagram UDP contenente una stringa di testo "*ciao*" da un host sender 137.200.70.14 (il client) a un host receiver 130.130.12.17 (il server), che ha come punto di accesso la porta 3069, stabilita tra quelle libere dal programmatore della applicazione.

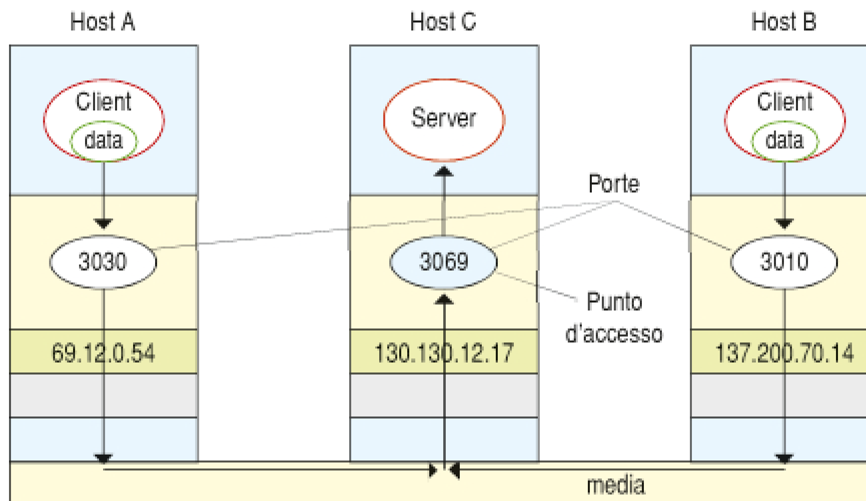
Il server manda in esecuzione l'applicazione e si mette in attesa sulla porta 3069 fino a che il client invia su di essa un datagram: è fondamentale che client e server si accordino sul numero della porta da usare e, naturalmente, il client deve conoscere l'indirizzo IP del server.



Indirizzo dei socket del server: <130.130.12.17:3069>

Al server arrivano segmenti che hanno differenti indirizzi IP sorgente e/o numeri di porta sorgente ma che possono avere, oltre che lo stesso indirizzo IP, anche lo stesso numero di porta destinazione: vengono consegnati allo stesso socket all'interno del terminale di destinazione.

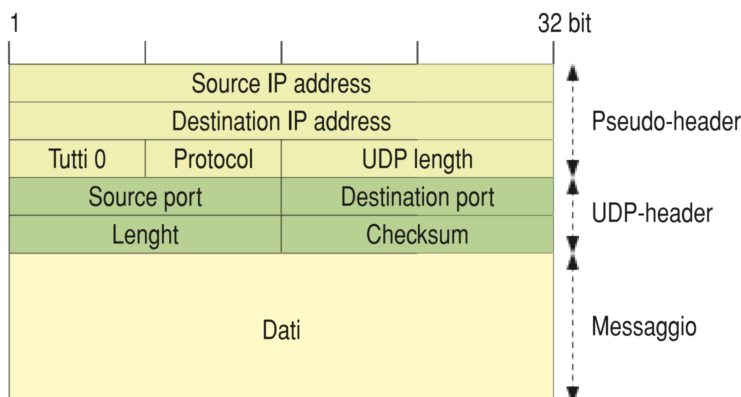
Quando il terminale di server riceve un segmento UDP, allora:
- legge il numero della porta di mittente;
- estrae il messaggio contenuto nel segmento;
- invia il messaggio al socket col numero di porta di destinazione specificato:
  - socket mittente host A: <69.12.0.54:3030>
  - socket destinatario host B: <137.200.70.14:3010>



Rilevazione degli errori

Il protocollo UDP è in grado di effettuare il controllo sui segmenti che trasferisce in modo da rivelare gli eventuali errori che si fossero generati nella trasmissione: ogni segmento viene analizzato alla sua ricezione e, nel caso in cui venga rivelato un errore, il segmento viene scartato oppure viene segnalata la presenza dell'errore all'applicazione che lo ha generato.

Per fare questo viene calcolato dal mittente del messaggio il checksum, analogamente a quanto accade per il checksum IP ma, come già detto prima, tenendo conto non solo dei campi del segmento UDP ma anche dello pseudo-header IP:



Il mittente genera il checksum considerando i byte del segmento come una sequenza di interi da 16 bit ed effettua la somma, cioè il complemento a 1, e memorizza il valore ottenuto nel campo checksum del segmento UDP. Il ricevente calcola il checksum del segmento ricevuto e controlla tale valore con il contenuto del campo checksum: se ci sono delle differenze rileva la presenza di un errore.



Zoom su...

ESEMPIO DI CALCOLO DEL CHECKSUM

Quando si sommano i numeri, un riporto dal bit più significativo deve essere sommato al risultato: vediamo con un esempio come operare per generare il checksum su due stringhe di 16 bit. Eseguiamo dapprima la somma dei bit in colonna

		1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
		1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
a capo	(1)	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1

e a questa è necessario aggiungere il riporto generato dalla overflow dei bit più significativi:

a capo	(1)	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
																	1
somma		1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0

a questo punto è possibile calcolare il checksum, complementando tutti i bit

somma	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1	1

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 L'acronimo UDP significa:

a) User Data Protocol
b) Uniform Datagram Protocol
c) User Datagram Protocol
d) Uniform Data Protocol

2 Quale tra le seguenti affermazioni è falsa per il protocollo UDP?

a) non garantisce la consegna
b) è non affidabile
c) non individua l'errore nei dati
d) è di tipo connectionless
e) non garantisce l'ordine di consegna

3 UDP ha i seguenti vantaggi (indicare quelli errati):

a) non introduce i ritardi temporali
b) non instaura una connessione tra i processi comunicanti
c) non viene richiesto che si mantenga la connessione
d) l'header di un segmento UDP è molto più corto
e) non attua controllo di congestione

4 Quali tra i seguenti campi non sono presenti nell'header UDP?

a) porta sorgente
b) porta destinatario
c) IP destinatario
d) lunghezza dati
e) checksum
f) dati

Test vero/falso

1 UDP utilizza l'IP per trasportare i messaggi. V F
2 UDP offre in più la capacità di distinguere tra più destinazioni all'interno di uno stesso host. V F
3 I segmenti inviati possono essere perduti durante la trasmissione pregiudicando il sistema. V F
4 In UDP avviene l'handshaking tra mittente e destinatario. V F
5 L'handshake è sempre necessario tra protocolli connection-oriented. V F
6 Il segmento UDP ha un header di non più di 8 byte. V F
7 Il valore di default Maximum Segment Size è di 536 byte. V F
8 Ogni segmento IP viene incapsulato in un datagram UDP. V F
9 Il socket del server è in grado di ricevere più richieste dallo stesso cliente. V F
10 Il protocollo UDP è in grado di effettuare il controllo su errori di indirizzamento. V F

Esercizio di collegamento

1 **Effettua il giusto accoppiamento tra le applicazioni che utilizzano UDP e i seguenti protocolli DNS, NFS, ntp, SNMP, Voip, RIP, tftp.**

Applicazione	Protocollo di strato applicativo
Telefonia via Internet	
Protocolli di instradamento	
Risoluzione di nomi	
Amministrazione di rete	
File server remoti	
Trivial File Transfer	
Network Time Protocol	

Esercizi di completamento

Il protocollo UDP è stato concepito per tutte quelle applicazioni per le quali non è necessaria una .... gestione delle connessioni, come per esempio le trasmissioni televisive .... dove può essere tranquillamente perso qualche .... senza pregiudicare la visione del programma.

Ogni segmento UDP fa uso di .... con un'intestazione di .... byte e può dare luogo a una modalità di trasferimento tra processi di tipo:

a) ....;
b) ....;
c) ....

Ogni .... UDP viene incapsulato in un ...., quindi la sua dimensione non può superare la dimensione massima della parte dati del .... ed è formato da due parti:

a) un ....;
b) un campo ...., di lunghezza variabile ma non superiore al ....

Il campo header è composto da 4 elementi:

a) ....;
b) ....;
c) ....;
d) ....

Il mittente del messaggio calcola il checksum, analogamente a quanto accade per il checksum IP ma tenendo conto non solo dei campi .... ma anche dello ....

Lezione 3

IL SERVIZIO DI TRASFERIMENTO AFFIDABILE

IN QUESTA LEZIONE IMPAREREMO...

- il concetto di segmento affidabile
- i meccanismi che realizzano un trasferimento affidabile
- il protocollo sliding window

Principi generali

Lo strato di rete implementa un servizio di trasferimento che non è affidabile in quanto è possibile che vengano persi dei segmenti o si generino degli errori: è compito dello strato di trasporto attuare meccanismi che permettano di eliminare i problemi presenti agli strati inferiori a esso.

Un servizio di trasferimento si dice affidabile se:
- tutti i messaggi sono consegnati a destinazione e giungono privi di errori;
- ciascun messaggio è consegnato una e una sola volta;
- i messaggi sono consegnati nello stesso ordine in cui sono stati trasmessi.

La trasmissione deve essere:
- priva di errori;
- senza di perdita di dati;
- senza duplicazioni nella consegna dei segmenti.

L'obiettivo principale del livello di trasporto è quello di offrire un servizio di trasferimento affidabile dei messaggi ai livelli superiori da affiancare al protocollo UDP. Introduciamo la definizione di un parametro che sarà utile nel seguito della trattazione: sappiamo che in una rete si definisce Round Trip Time (RTT) l'intervallo di tempo che intercorre tra l'istante in cui un segmento inizia a essere trasmesso dalla sorgente e l'istante in cui la sorgente ne riceve il relativo messaggio di riscontro, cioè il tempo di andata e ritorno tra mittente e ricevente. A questo parametro viene generalmente affiancato un altro parametro chiamato ◀ Retransmission Time Out (RTO). ▶



◀ Retransmission Time Out (RTO) Indica il massimo intervallo di tempo che può intercorrere tra l'istante di trasmissione di un segmento e l'istante di ricezione del corrispondente riscontro prima che la sorgente consideri perso il segmento stesso. ▶



◀ **Retransmission Time Out (RTO)** TCP retransmission timeout (RTO) occurs when ACK does not arrives on time, or does not arrive at all! Usually, RTO is not a fixed value, but changes to gain better network performance. ▶

Se il mittente non riceve il segnale di ricezione (ACK) entro un fissato tempo limite RTO avviene la ritrasmissione dei segmenti da parte del mittente.

Il dimensionamento del timeout è un aspetto critico nelle prestazioni del protocollo:
- valore troppo piccolo: potrebbero essere considerati persi alcuni segmenti in ritardo a causa di congestione e venire ritrasmessi con conseguente perdita di efficienza;
- valore troppo grande: verrebbe ritrasmesso con notevole ritardo un segmento considerato perso introducendo quindi rallentamenti e conseguente perdita di efficienza.

In questa lezione verranno descritti i meccanismi implementati dallo strato di trasporto e utilizzati nel protocollo TCP che permettono a due socket di scambiare segmenti in modo affidabile mediante un canale logico.

I meccanismi impiegati

Per rendere affidabile un canale è necessario che venga effettuata la rivelazione degli errori in ricezione e quindi avvenga la ritrasmissione dei segmenti persi o quelli in cui sono stati rivelati errori; i meccanismi impiegati per realizzare un trasferimento di segmenti affidabile possono essere riassunti in:
- numerazione dei segmenti trasmessi e trasmissione di messaggi di riscontro con numero di sequenza;
- impiego di un temporizzatore (timer) in trasmissione;
- impiego di finestre in trasmissione e in ricezione.

Numerazione dei segmenti trasmessi

Un segmento è composto da più byte e lo strato di trasporto considera la sequenza dei segmenti trasmessi come una sequenza ordinata di byte: viene quindi numerata progressivamente. Il numero di sequenza (sequence number) associato a un segmento è il numero d'ordine del primo byte che compone il segmento stesso e serve a posizionare il carico utile del segmento TCP all'interno del flusso di dati.

Il campo Sequence Number (SN) contiene il numero sequenziale di ciascun byte di dati a partire dall'◀ Initial Sequence Number (ISN) ▶: il primo vero byte di dati spedito ha come numero di sequenza SN=ISN+1, che solitamente ha valore 1. Quindi il SN identifica la posizione nello stream del primo byte di dati del segmento: se per esempio il primo byte nel flusso di dati ha come SN=1 (quindi il ISN aveva valore ISN=0) e sono già stati trasferiti 5000 byte, il primo byte di dati nel segmento corrente è 5001 e quindi il suo SN viene settato a 5001.



◀ Initial Sequence Number (ISN) Si tratta di un numero intero progressivo che viene inizializzato a un valore compreso tra 0 e $(2^{32} - 1)$ mediante un sistema di numerazione di tipo ciclico, ovvero: ISN = x mod 2^{32}. Per esempio per determinare il numero di sequenza ISN del prossimo segmento a partire dal valore dell'ultimo segmento N compreso tra N = $\{0,\ldots, (2^{32} - 1)\}$ e dalla sua dimensione K, si sostituisce a il valore x = N + K ottenendo ISN = (N + K) mod 2^{32}. ▶

Lo standard TCP non richiede che ogni sistema inizi a numerare i byte partendo da uno specifico numero; ogni sistema sceglie liberamente il numero da cui iniziare la numerazione. Inoltre si aspetta di ricevere il segmento successivo all'ultimo segmento ricevuto in ordine, ovvero quello il cui numero di sequenza è pari al numero di sequenza dell'ultimo segmento ricevuto in ordine più la dimensione

del carico utile dello stesso segmento (cioè del suo campo Data). Quando viene ricevuto un segmento, si controlla se il numero di sequenza ricevuto è quello atteso e in caso affermativo il destinatario può inviare direttamente il carico utile al processo di livello applicativo e liberare i propri buffer di ricezione: se invece riceve un numero di sequenza maggiore di quello atteso, memorizza temporaneamente i dati nel buffer di ricezione nell'attesa che giungano i segmenti mancanti andati persi oppure in ritardo sulla rete. Se il numero di sequenza ricevuto è inferiore a quello atteso significa che questo segmento è già stato ricevuto e quindi si tratta di un duplicato che viene scartato.

In ogni segmento inviato è presente un Acknowledgment Number, o numero di riscontro, che *riguarda il flusso di dati nella direzione opposta*: il numero che il mittente invia al destinatario è il numero del primo byte che il mittente si attende di ricevere e quindi il numero che deve avere il dato in risposta a quella trasmissione.

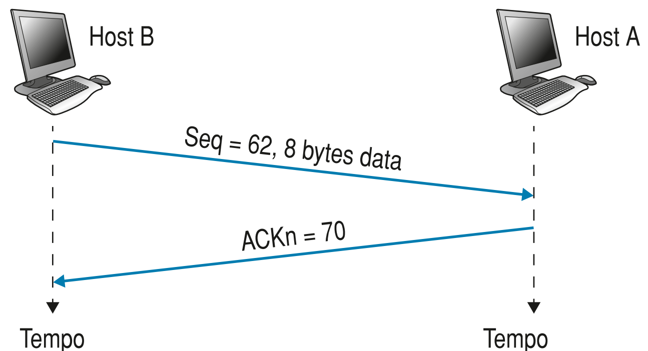
Il protocollo TCP adotta la politica di conferma cumulativa: il numero di riscontro che viene trasmesso indica al ricevente che il mittente ha ricevuto e inoltrato al processo applicativo di livello superiore il segmento avente numero di sequenza uguale al numero di riscontro indicato (-1) e anche *tutti i segmenti a esso precedenti*.

In trasmissione, il protocollo TCP mantiene temporaneamente una copia di tutti i dati inviati ma non ancora riscontrati e solo al momento del riscontro tramite l'ACKn libera il proprio buffer da questi dati dato che ha avuto la conferma di consegna al destinatario. Questa tecnica è nota come tecnica Go-Back-N.

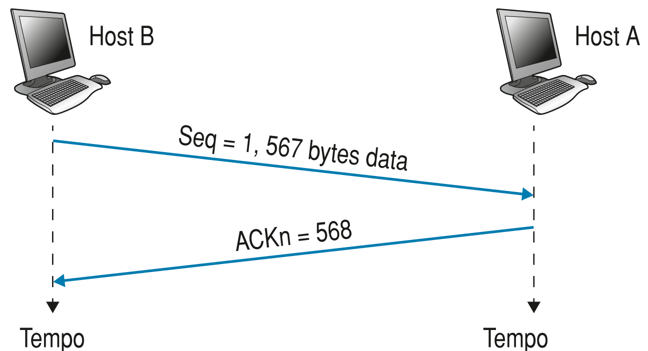
Quindi il segmento di acknowledgment svolge due funzioni, cioè quella di segnalare l'avvenuta ricezione (positive acknoledgment) e quella di controllare la sequenza e quindi l'ordine dei segmenti (flow control).

ESEMPIO 7

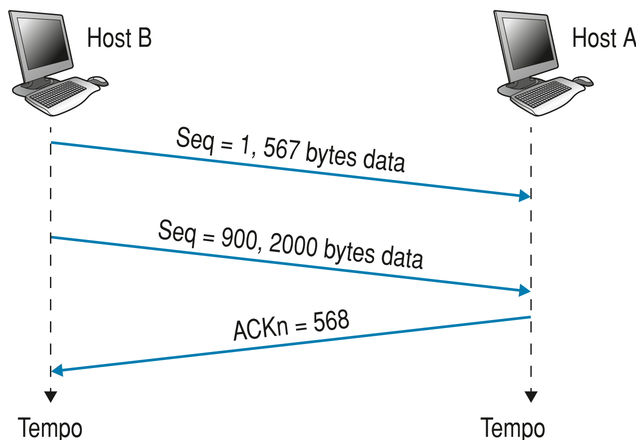
Per esempio, se il primo byte spedito ha ISN=62 e sono stati trasferiti correttamente 8 byte, l'Acknowledgment Number è 70. ▶



Se il primo byte spedito ha ISN=1 e sono stati trasferiti correttamente 567 byte, l'Acknowledgment Number è 568. ▶



La somma dei byte ricevuti deve naturalmente rispettare la sequenza, altrimenti il ricevente non li conteggia e non viene aggiornato il totale presente nel contatore: se l'host A ha ricevuto dall'host B un segmento con i byte da 1 a 567 e un segmento con i byte da 900 a 2000 riconosce che "ne manca un pezzo" e come conferma all'host A invia un segmento dove scrive nel campo riscontro il valore 568 che è il numero di byte che si aspetta dall'host B per ricreare il flusso completo. ▶



Non è richiesta la trasmissione di un messaggio di ACK a ogni segmento ricevuto: il riscontro che arriva alla sorgente è di tipo cumulativo.

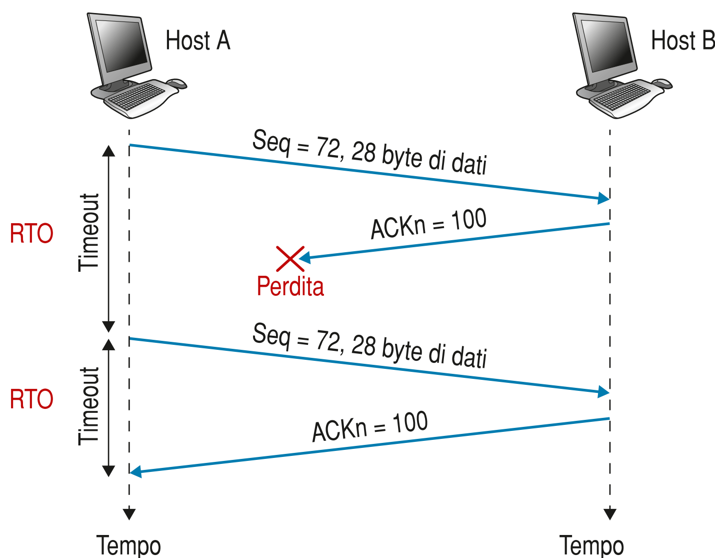
Con NACK si intendono i messaggi di ACK negativi, cioè i messaggi che indicano un riscontro negativo: i protocolli di trasporto non prevedono l'impiego di tali messaggi, in quanto al loro posto si utilizzano i messaggi NACK di tipo implicito, deducendo dai valori di ACK il comportamento da intraprendere.

Temporizzazione della trasmissione

Timer di ritrasmissione

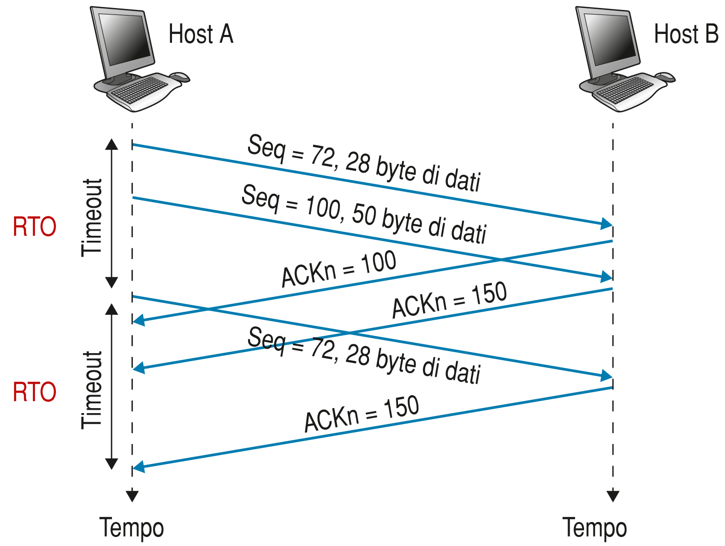
Per ciascun segmento inviato, TCP avvia un timer, detto *timer di ritrasmissione* RTO (Retransmission Time Out, generalmente indicato come timeout) che indica al modulo di origine dopo quanto tempo deve essersi considerato perso l'ultimo segmento trasmesso e quindi necessita la sua trasmissione: se nessun ACK viene ricevuto dal mittente in quell'intervallo di tempo questo ritrasmette tutti i segmenti inviati dall'ultimo che è stato confermato.

Nel caso riportato in figura viene perso il messaggio di ACK e quindi, al termine del RTO, l'host A provvede a ritrasmettere il segmento precedente. ▶



Nell'esempio seguente il segmento di ACK non raggiunge la destinazione nell'intervallo di tempo RTO e quindi nessuno dei due segmenti inviati dall'host A viene confermato: si procede con la ritrasmissione del primo segmento. ▶

A questo punto si viene a creare un problema di duplicati: 2 segmenti uguali raggiungono l'host B che però ha memorizzato il numero di sequenza e quindi è in grado di ignorare la seconda trasmissione e richiedere il segmento mancante, cioè quello successivo ai primi due ricevuti (nel nostro caso il 150).



La corretta impostazione di questo timer è difficile ma molto importante, in quanto un timer troppo breve comporta ritrasmissioni inutili (il timer scatta mentre il riscontro o il pacchetto sono ancora in viaggio), mentre un timer troppo lungo comporta attese in caso di perdita di pacchetti.

Timer di keepalive

Una ulteriore temporizzazione viene effettuata mediante il timer di keepalive: inizia il conteggio alla ricezione di ogni pacchetto e quando scade dichiara la connessione caduta per eccessiva inattività. Quindi questo timer interviene per individuare le situazioni in cui cade la connessione.

> Per evitare che il timer scada inopportunamente, il TCP si occupa di mandare dei pacchetti vuoti se il mittente non ha nulla da inviare.

Timed wait

Il timed wait è invece il tempo che viene atteso prima di disconnettere effettivamente una connessione ed è pari al doppio del tempo di vita di un comune pacchetto: questo evita che dei pacchetti possano rimanere circolanti per la rete anche dopo la chiusura.

Timer di persistenza

Come descritto successivamente, il TCP utilizza il metodo della finestra scorrevole per gestire il flusso di dati che il ricevente è in grado di accettare, e tra i valori validi di questo campo vi è anche lo zero, a significare che il ricevente richiede l'interruzione momentanea dell'invio di dati.
Se viene perso proprio il pacchetto che riapre la finestra, il mittente del canale TCP rimarrà però in attesa indefinita: per evitare questa situazione il TCP avvia un timer, detto timer di persistenza, ogni qual volta il ricevente chiude la finestra.

Finestra di trasmissione e ricezione

Per effettuare la gestione dei segmenti inviati e ricevuti, e quindi per poter individuare quali segmenti devono eventualmente essere ritrasmessi, il terminale mittente utilizza quella che prende il nome di ◀ finestra di trasmissione. ▶

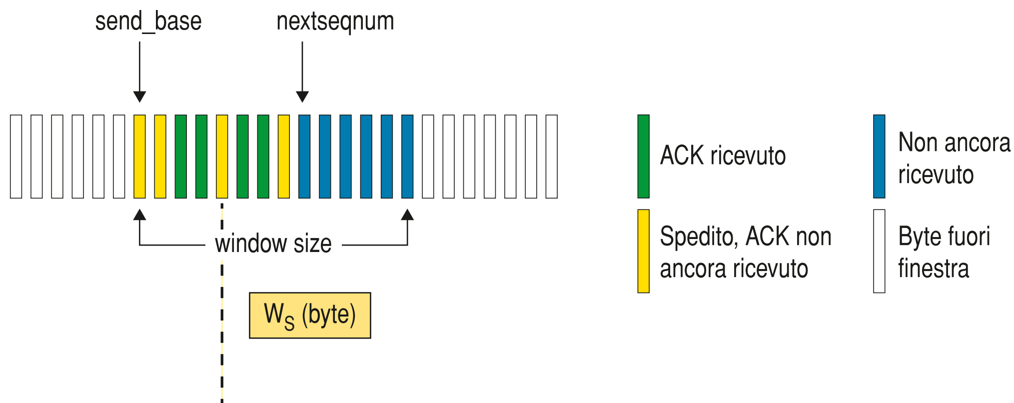


◀ **Finestra di trasmissione** La finestra di trasmissione ha dimensione W_s e in essa vengono memorizzati tutti i numeri di sequenza dei byte che la sorgente può trasmettere senza bisogno di ricevere alcun messaggio ACK da parte della destinazione. ▶

La finestra di trasmissione viene gestita come una struttura a coda, utilizzando 2 variabili (2 puntatori):

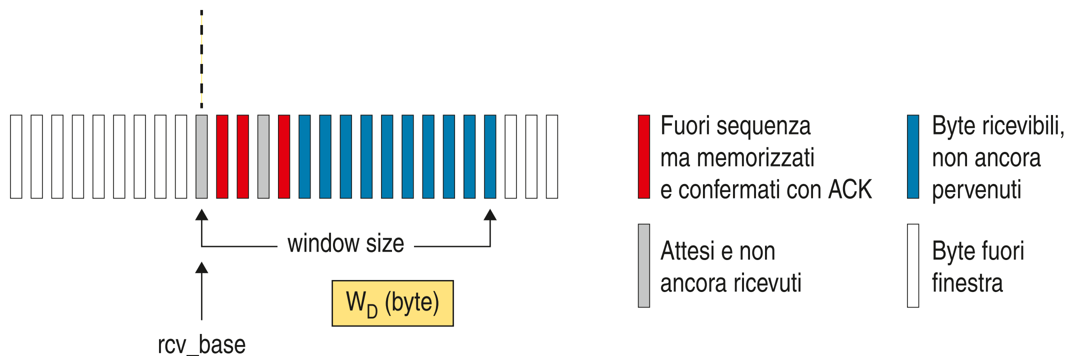
- sendbase: rappresenta il numero d'ordine del byte più vecchio tra quelli trasmessi ma non ancora riscontrati positivamente dalla destinazione;
- nextseqnum: rappresenta il numero d'ordine del prossimo byte che cade nella finestra corrente e che deve ancora essere trasmesso per la prima volta.

Sendbase individua l'estremo inferiore della finestra di trasmissione corrente e nextseqnum un elemento centrale, mentre l'estremo superiore è dato dalla somma tra sendbase e la larghezza W_s.



Alla ricezione di un ACK il mittente aggiorna il valore di sendbase con il valore che gli viene comunicato se tale valore risulta essere superiore al valore corrente, altrimenti lo lascia inalterato. Aggiornando il valore inferiore la finestra viene spostata collocandosi più a destra dello stesso numero di byte.

La finestra di ricezione, avente dimensioni W_D pari al numero di byte che il ricevente è in grado di ricevere in quel momento, è una struttura di dati che viene mantenuta aggiornata nell'host di destinazione.



Nella finestra di ricezione vengono memorizzati i numeri d'ordine dei byte che la destinazione è disposta a ricevere consecutivamente, prima di dover inviare alcun messaggio ACK alla sorgente. La gestione di questa finestra viene effettuata con un'unica variabile, rcv_base, che contiene il minimo valore che il destinatario si attende sia perché quel byte non lo ha ancora ricevuto oppure perché lo ha ricevuto errato.

Se per esempio viene ricevuto un segmento con numero di sequenza maggiore, se è privo di errori e cade nella finestra di ricezione viene memorizzato ma non si aggiorna il rcvbase e si rimane in attesa della parte di messaggio antecedente che ancora non è pervenuta.

La dimensione delle due finestre generalmente è diversa in quanto le velocità di elaborazione di mittente e destinatario possono essere anche molto diverse: affinché il mittente non saturi il destinatario, è necessario che questo invii al primo la dimensione della memoria di ricezione in modo che il mittente si possa regolare modificando i flussi di trasmissione: vedremo come nel protocollo TCP il ricevente comunica al trasmittente la dimensione opportuna della finestra attraverso il campo Window presente nell'header dei segmenti TCP.

Il meccanismo descritto prende il nome di finestra scorrevole o ◀ **sliding window** ▶ protocol.



◀ **Sliding window** A sliding window protocol is a feature of packet-based data transmission protocols.
The "window" is the maximum amount of data we can send without having to wait for ACKs. In summary, the operation of the algorithm is as follows:

- transmit all the new segments in the window;
- wait for acknowledgement/s to come (several packets can be acknowledged in the same ACK);
- slide the window to the indicated position and set the window size to the value advertised in the acknowledgement.

When we wait for an acknowledgement to a packet for some time and it has not arrived yet, the packet is retransmitted. When the acknowlegement arrives, it causes the window to be repositioned and the transmission continues from the packet following the one transmitted last. ▶

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Il servizio di trasferimento è detto affidabile se (indica la voce non corretta):
- a) tutti i messaggi sono consegnati a destinazione
- b) tutti gli errori vengono individuati
- c) ciascun messaggio è consegnato una e una sola volta
- d) i messaggi sono consegnati nello stesso ordine in cui sono stati trasmessi
- e) tutti i messaggi giungono privi di errori

2 Con TPDU si intende:
- a) Transport Data Protocol User
- b) Transport Device Protocol User
- c) Transport Device Protocol Unit
- d) Transport Data Protocol Unit

3 L'acronimo RTO sta a indicare:
- a) Rete Time Out
- b) Roundtrip Time Out
- c) Retransmission Time Out
- d) Release Time Out

4 Quale tra i seguenti non è un meccanismo introdotto per rendere affidabile un canale?
- a) numerazione dei segmenti trasmessi
- b) trasmissione di messaggi di riscontro con numero di sequenza
- c) ritrasmissione di messaggi errati
- d) impiego di un temporizzatore in trasmissione
- e) impiego di finestre in trasmissione
- f) impiego di finestre in ricezione

5 Se un Host riceve i segmenti con valore (seq 100-10, seq 130-10, seq 110-20) che valore conferma?
- a) 110
- b) 111
- c) 120
- d) 121
- e) 130
- f) 131
- g) 140
- h) 141

6 Quale di queste affermazioni è errata in riferimento alle finestra di trasmissione?
- a) W_s indica il numero di byte che la sorgente può trasmettere senza bisogno di ricevere ACK
- b) W_s può essere di dimensioni variabili a seconda dell'host ricevente
- c) sendbase rappresenta il numero d'ordine del byte più vecchio tra quelli trasmessi
- d) nextseqnum rappresenta il numero d'ordine del prossimo byte non ancora trasmesso
- e) sendbase individua l'estremo inferiore della finestra di trasmissione corrente
- f) nextseqnum individua l'estremo superiore della finestra di trasmissione corrente

Test vero/falso

1. Il RTT è sempre maggiore o uguale al RTO. V F
2. Un valore troppo grande di RTO può generare un aumento della congestione. V F
3. Per rendere affidabile un canale si impiegano finestre in trasmissione e in ricezione. V F
4. L'ISN è un intero che viene inizializzato a un valore compreso tra 0 e (223-1). V F
5. Il primo vero byte di dati spedito ha come numero di sequenza SN=ISN+1. V F
6. Il valore dell'Acknowledgment Number contiene il numero di sequenza dell'ultimo byte ricevuto. V F
7. È richiesta la trasmissione di un messaggio di ACK a ogni segmento ricevuto. V F
8. Con NACK si intendono i messaggi di ACK negativi obbligatori in caso di mancanza di ricezione. V F

Lezione 4

IL PROTOCOLLO TCP

IN QUESTA LEZIONE IMPAREREMO...

- le caratteristiche del protocollo TCP
- la struttura del segmento TCP
- a stimare il valore del timeout

Il protocollo TCP

La caratteristica fondamentale del protocollo TCP è quella di essere un ◀ protocollo punto-punto ▶, cioè orientato alla connessione tra due host, chiamati client e server.

La connessione avviene mediante un ◀ handshaking ▶ con lo scambio di messaggi di controllo che inizializza lo stato del mittente e del destinatario prima di iniziare a scambiare i dati.

Prima di iniziare la connessione vera e propria tra due computer si crea una connessione di tipo handshake che consiste nella trasmissione dei pacchetti necessari per regolare i parametri di connessione.



◀ Protocollo punto-punto I protocolli punto-punto consentono un collegamento a basso costo per mezzo di linee seriali o linee telefoniche commutate. ▶



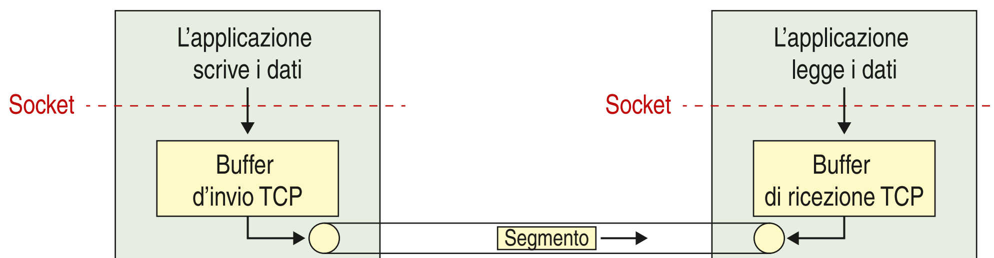
◀ Handshaking Mediante questa tecnica due elementi stabiliscono regole hardware o software comuni, ovvero la velocità, i protocolli di compressione, di criptazione, di controllo degli errori ecc. ▶

Il suono prodotto dal modem nella fase di connessione a Internet viene anche chiamato handshake.

Successivamente avviene la comunicazione vera e propria nella quale il flusso dei dati è affidabile, in sequenza, bidirezionale, ◀ full duplex ▶, e viene realizzato mediante un buffer d'invio e uno di ricezione.

◀ **Full duplex** Full-duplex data transmission means that data can be transmitted in both directions on a signal carrier at the same time. ▶





Le creazione del canale di comunicazione avviene mediante la connessione tramite socket (come abbiamo visto in precedenza utilizzando il protocollo UDP): il server può effettuare contemporaneamente più connessioni ma, naturalmente, affinché una connessione possa essere distinta dalle altre, ciascuna di esse non può avere la stessa coppia di socket.

La tabella seguente riporta la numerazione di alcune well-known ports di classiche applicazioni TCP.

Porta	Protocollo	Applicazione
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	Email
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
119	NNTP	USENET news
110	POP-3	Remote email access

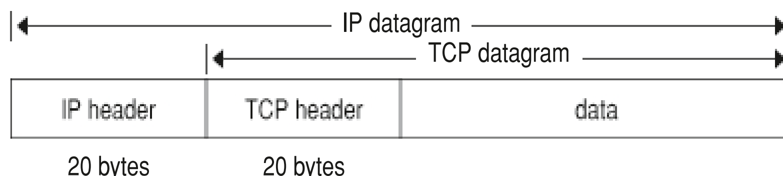
Le porte con numero inferiore a 1024 sono porte note o riservate (well-known ports) mentre per tutte le altre viene attivato un unico processo (programma ◀ inetd ▶) che intercetta le richieste di connessione, crea il processo relativo e ridirige la connessione da se stesso al processo appena creato, liberandosi per intercettare future richieste.



◀ Inetd È anche chiamato Internet Super-Server in quanto gestisce le connessioni necessarie per i servizi richiesti: quando arriva una richiesta di connessione, inetd determina a quale programma la connessione è destinata, esegue quel particolare processo e affida a lui il socket. ▶

Il segmento TCP

TCP suddivide i dati dell'utente in segmenti di non più di 64 KB e li incapsula in datagrammi IP, come già descritto per il datagramma UDP.



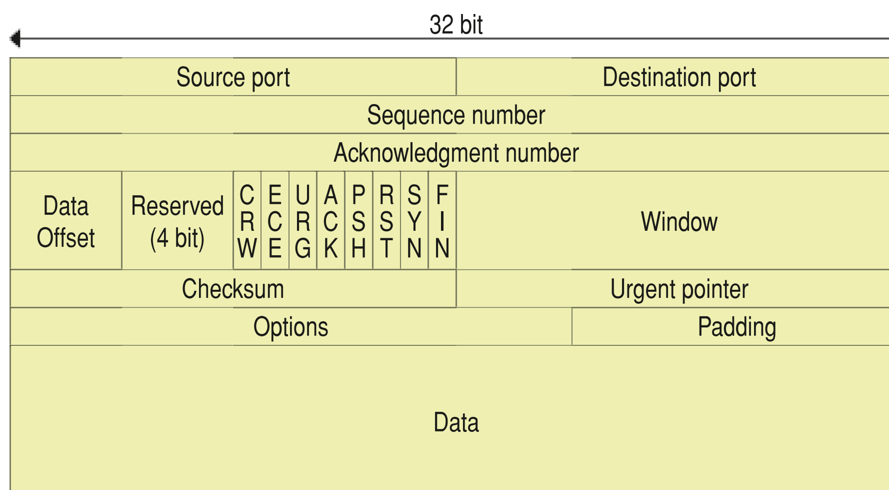
I segmenti sono organizzati in due sezioni:
- un'intestazione standard di 20 byte;
- un payload di dimensione variabile (anche nulla ma di dimensione massima 64 Kb) contenente i dati di applicazione.

Il mittente controlla il flusso dei dati trasmesso in modo da non sovraccaricare il destinatario: viene anche definito un parametro che indica la dimensione massima di segmento (Maximum Segment Size MSS). La corrispondente massima dimensione del blocco dati di applicazione che può essere contenuto nel segmento non deve essere superiore alla dimensione del payload IP dedotta dalla occupazione dell'header IP e dell'header TCP (per esempio, 65535 – 20 – 20 = 65495 byte); deve inoltre rispettare i limiti imposti alle dimensioni dei pacchetti dalle reti che deve attraversare (Maximum Trasmission Unit MTU) e quindi possiamo esprimere la relazione tra MTU e MSS in:

$$MSS = MTU - 20 - 20$$

In generale non è possibile conoscere la MTU di ogni rete intermedia che verrà attraversata dai segmenti e viene utilizzato un algoritmo detto Path MTU discovery (RFC 1191) per la definizione del suo valore: un tipico valore per MTU sono i 1500 byte imposti da Ethernet.

La struttura dei segmenti TCP è la seguente:



Descriviamo i singoli campi dell'intestazione TCP:

- Source/Destination port: numero di porta sorgente e destinazione;
- Sequence number: numero di sequenza del primo byte contenuto nel segmento; è il numero di sequenza iniziale su cui sincronizzarsi se è presente il bit SYN=1;
- Acknowledgment number: se il bit ACK è a 1, questo è il numero di sequenza del blocco di dati che ci si aspetta di ricevere;
- Data offset: numero di parole di 32 bit dell'intestazione TCP; indica dove iniziano i dati;
- Reserved: 4 bit riservati per uso futuro; devono essere posti a zero;
- Control bit: sono 8 flag
  - URG posto a 1 se si deve considerare il campo Urgent Pointer;
  - ACK posto a 1 se si deve considerare il campo Acknowledgment Number;
  - PSH posto a 1 indica la funzione di push, per la consegna immediata delle informazioni;
  - CWR (Congestion Window Reduced) posto a 1 quando l'host sorgente ha ridotto la velocità di trasmissione per ridurre la congestione (RCF 3168);
  - ECE (ECN echo) posto a 1 se l'host supporta l'◀ Explicit Congestion Notification ▶ ECN;
  - RST posto a 1 per resettare la connessione e rifiutare un segmento o un tentativo di connessione non validi;
  - SYN posto a 1 per stabilire la connessione e per sincronizzare i numeri di sequenza;
  - FIN posto a 1 per indicare la fine dei dati da trasmettere e chiudere la connessione in una direzione.

Gli ultimi tre bit sono utilizzati per impostare e chiudere la connessione.

- Window: dimensione della finestra in ricezione sliding window per il controllo di flusso, cioè il numero di byte che il ricevitore è disposto a ricevere a partire dal numero di sequenza contenuto nel campo Acknowledgment Number;
- Checksum: controllo d'errore su intestazione e dati che viene effettuato su blocchi da 16 bit.



◀ Explicit Congestion Notification La congestione dei pacchetti si verifica quando il router riceve pacchetti a una velocità superiore a quella possibile sul link su cui dovrebbe spedirli. Solitamente vengono parcheggiati nella memoria sino alla completa saturazione. Per evitare l'inconveniente solitamente i router compiono due azioni:

1. abbassano la velocità di trasmissione;
2. ripetono i pacchetti scartati.

La tecnica ECN (Explicit Congestion Notification) evita la congestione del router notificando esplicitamente che si sta congestionando utilizzando un apposito campo (il campo ECN) presente nell' header IP. Quando il router rileva uno stato di "blanda" congestione, i bit di tale campo vengono impostati in una configurazione detta Congestion Experienced CE. Il campo ECN è costituito dai bit 6 e 7 del campo TOS. ▶

La connessione TCP

Abbiamo detto che la connessione/sconnessione TCP avviene attraverso una fase preliminare chiamata handshaking tra client e server che permette di stabilire un canale dedicato tra loro e alla fine di rilasciarlo: vediamo la procedura di connessione e successivamente quella di rilascio del canale e chiusura della connessione.

Nella letteratura tecnica viene spesso indicato con ACK nei diagrammi temporali sia il flag di stato che il campo Acknowledgment Number lasciando l'interpretazione al lettore in base al valore contenuto e al suo contesto di utilizzo: noi, per evitare ambiguità, indicheremo con ACK il flag e con ACKn l'Acknowledgment Number.
Inoltre, quando viene indicato sul diagramma un flag, generalmente si sottintende che il suo valore è settato a 1: noi, per completezza, lo scriveremo in modo esplicito.

Apertura delle connessioni

La procedura utilizzata per instaurare in modo affidabile una connessione TCP tra due host è chiamata three-way handshake (stretta di mano a 3 vie), indicando la necessità di scambiare 3 messaggi tra host mittente e host ricevente affinché la connessione sia creata correttamente.

Fu proposta nel 1975 da Tomlinson, il progettista informatico che inventò l'◀ **email** ▶, per risolvere il problema dei duplicati relativi alla fase di attivazione della connessione, che descriveremo nella prossima lezione.

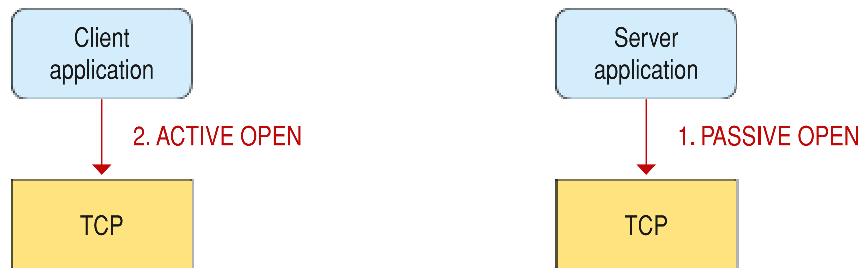


◀ **email** Raymond Samuel Tomlinson is a US programmer who implemented an email system in 1971 on the ARPANET. It was the first system able to send mail between users on different hosts connected to the ARPAnet. To achieve this, he used the @ sign to separate the user from their machine, which has been used in email addresses ever since. ▶

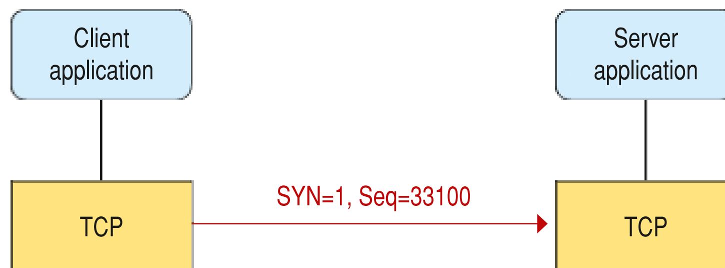
La seguente procedura illustra come avviare la connessione, nella quale è necessario che sui due host sia presente il software specifico per il servizio desiderato e che venga mandato in esecuzione.

1. Il server manda in esecuzione l'applicazione e rimane in attesa passiva, sulla porta specifica per essa riservata, di una richiesta di connessione (Passive Open);

2. quando un client vuole comunicare con un server manda in esecuzione l'applicativo specifico che conosce l'indirizzo IP del server e il numero di porta riservato per il servizio desiderato (indirizzo del socket): viene quindi inviata una richiesta TCP (Active Open);

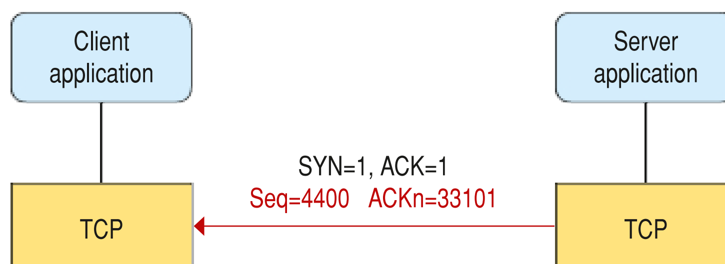


3. il client TCP genera in modo casuale un numero di sequenza iniziale (per esempio Seq=33100) e manda un messaggio di SYNchronize (flag SYN=1) contenente questo numero di sequenza (il flag ACK=0);

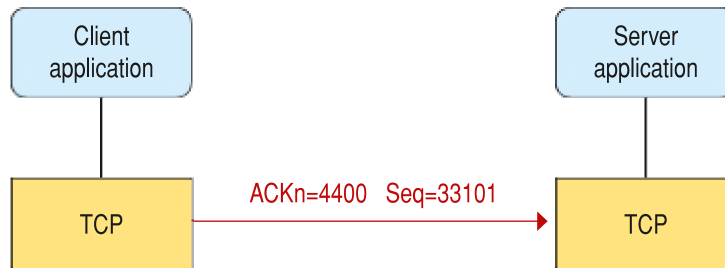


In questa prima trasmissione non vengono inviati dati ma viene inviata solo la numerazione del segmento affinché si possa perfezionare l'operazione di setup.

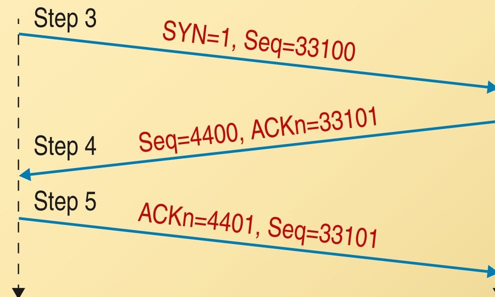
4. Alla ricezione del segmento con SYN=1, il server genera casualmente un suo numero di sequenza iniziale (per esempio Seq=4400) e risponde con un segmento avente i flag SYN=1, flag ACK=1 (comunemente chiamato segmento SYN/ACK), che contiene in risposta l'acknowledgment number uguale a ACKn=33101, incrementando cioè di 1 il valore ricevuto dal TCP client come conferma di ricezione.



5 Alla ricezione del SYN/ACK inviato dal server, il client risponde con un ACK di conferma ricezione incrementando di 1 il corrispondente valore (ACKn=4401) e inizia a inviare i primi dati nel payload indicando il numero di sequenza del primo byte, cioè Seq=33101.



Sintetizziamo con la seguente figura le operazioni della procedura di three-way handshaking.



6 7 Ora il TCP client e il TCP server comunicano alla applicazione che la connessione è aperta e le due applicazioni possono scambiarsi i dati nel canale logico che si è creato.



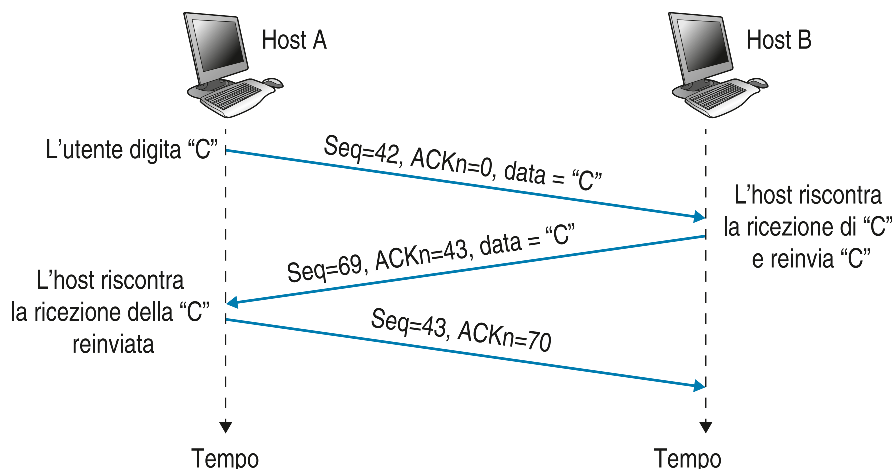
ESEMPIO 8

Vediamo ora un esempio di comunicazione in TCP per meglio comprendere l'utilizzo dei registri Sequence Number (Seq) e Acknowledgment Number (ACKn).

Ogni host li utilizza in modo indipendente, registrando in essi
- **A** Seq: "numero" del primo byte nel flusso di byte che sta trasmettendo;
- **B** ACKn: contiene il numero di sequenza del prossimo byte che si aspetta di ricevere.

Nel nostro esempio abbiamo una semplice applicazione dove il ricevente risponde come echo lo stesso carattere inviato dal mittente; i numeri di sequenza iniziali sono generati random e valgono:
- host A => Seq=42
- host B => Seq=69

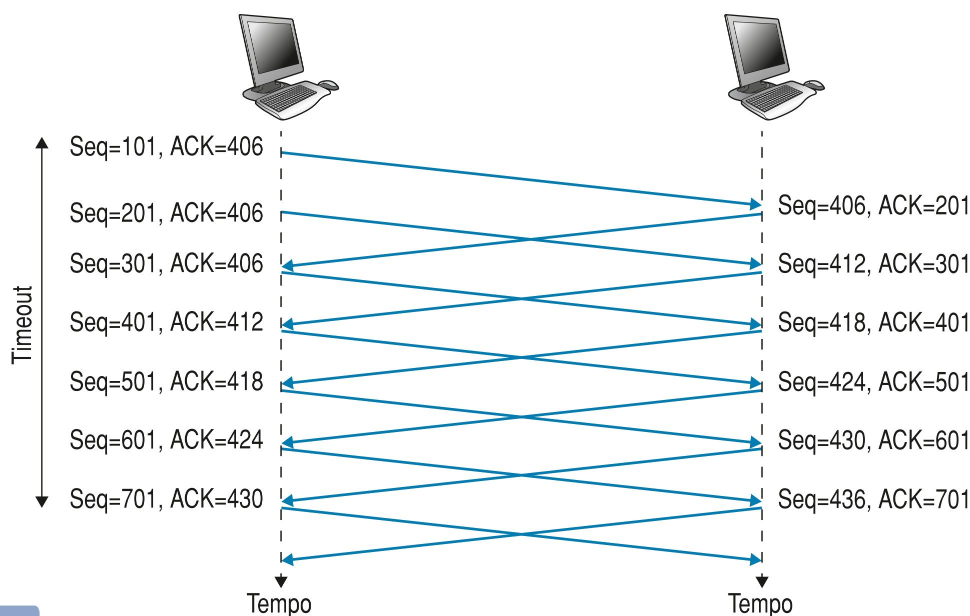
Alla trasmissione del primo carattere il valore di ACKn per l'host A è uguale a 0.



Alla ricezione del dato, il byte numero 42 indicato dal valore di Seq dell'host A, l'host B risponde con la conferma di ricezione inviando lo stesso carattere ma con il proprio valore di Seq=69 e incrementando nell'ACKn il numero di byte ricevuti di 1, in modo da indicare al mittente che il prossimo byte che si aspetta di ricevere è il numero 43.
A sua volta l'host A, non appena riceve il messaggio da B, gli risponde confermando il numero di sequenza ricevuto (Seq=43) che è il numero del byte che si aspetta di ricevere (inoltre aggiorna ACKn=70, dato che ha ricevuto il byte numero 69).

ESEMPIO 9

Vediamo un ultimo esempio, dove ipotizziamo che il primo host invii segmenti lunghi 100 byte ciascuno. Questo host riscontra i segmenti inviati dall'altro, e inoltre invia dati lunghi 6 byte ogni volta.



Chiusura della connessione

La connessione TCP tra due host non è considerata una singola connessione bidirezionale ma piuttosto una coppia di connessioni monodirezionali: quando si vuole chiudere un canale è quindi necessario che venga chiusa la trasmissione in entrambi i sensi.
Può infatti verificarsi la situazione di *connessioni aperte a metà*, in cui solo uno dei due terminali ha chiuso la connessione e non può più trasmettere, ma può (e deve) ricevere i dati dall'altro terminale.

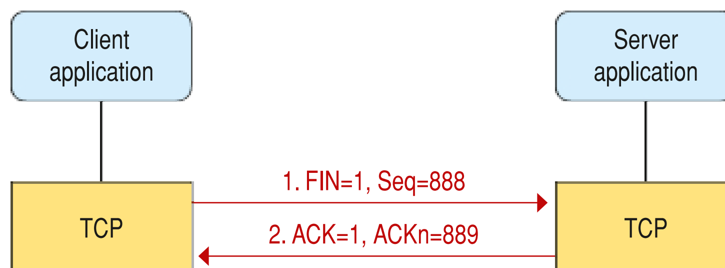
È quindi possibile avere due modalità di chiusura della connessione: con un handshake a tre vie, in cui le due parti chiudono contemporaneamente le rispettive connessioni, o con uno a quattro vie, in cui le due connessioni vengono chiuse in tempi diversi:

- l'handshake a 3 vie è omologo a quello usato per l'apertura della connessione, con la differenza che il flag utilizzato è il FIN invece del SYN. Un terminale invia un pacchetto con la richiesta FIN, l'altro risponde con un FIN + ACK, infine il primo manda l'ultimo ACK e l'intera connessione viene terminata;
- l'handshake a 4 vie invece viene utilizzato quando la disconnessione non è contemporanea tra i due terminali in comunicazione. In questo caso uno dei due terminali invia la richiesta di FIN e attende l'ACK. L'altro terminale farà poi altrettanto, generando quindi un totale di 4 pacchetti.

Vediamo nel dettaglio l'handshake a 4 vie:

1. Il client inizia la procedura di chiusura della connessione inviando un messaggio contenente gli ultimi dati che deve trasmettere e settando a 1 il flag di FIN (flag FIN=1).

2. Il server come prima operazione invia un messaggio di ACK per confermare la ricezione dei dati.

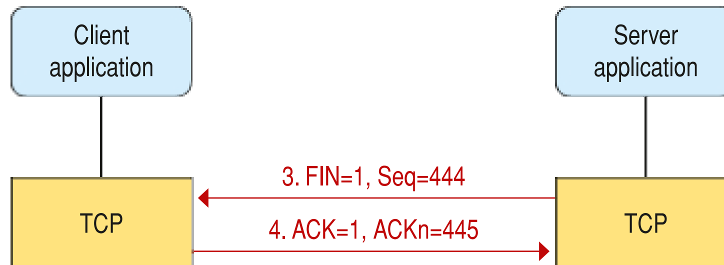


In questo momento la comunicazione nella direzione **server->client** è aperta e quindi possono essere trasferiti altri pacchetti verso il **client**: di ciascuno il **client** invierà il corrispondente messaggio di ACK al server.

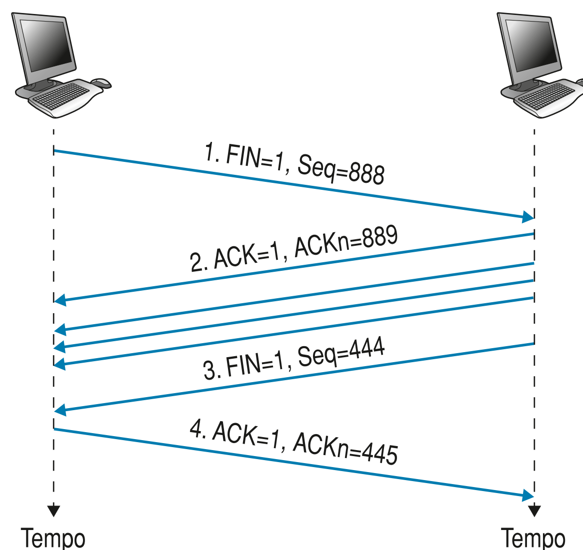


3. Quando anche il **server** decide di chiudere la connessione invia un messaggio con il flag FIN settato al valore 1…

4 … che viene confermato con un ACK finale dal client che aveva già chiuso la connessione in precedenza.



In sintesi possiamo rappresentare quanto descritto nel seguente diagramma temporale:



Stima e impostazione del timeout

Come è possibile intuire dalla analisi degli scambi di messaggi tra client e server un ruolo fondamentale nel protocollo TCP lo ha il dimensionamento del timeout dato che:

- se è troppo breve il mittente immetterà molteplici ritrasmissioni dello stesso segmento in quanto potrebbe non arrivarne la conferma in tempo utile;
- se è troppo lungo viene rallentato il recupero dei segmenti persi.

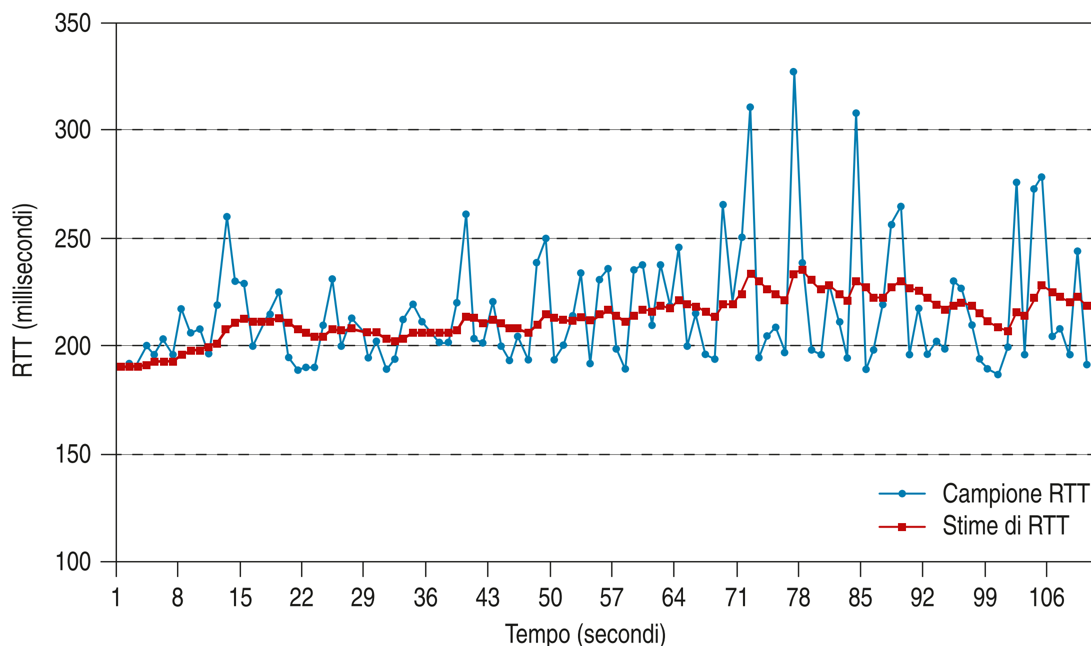
> Il valore ottimale del timeout non è un parametro standard ma dipende fortemente dal ritardo della rete e deve essere determinato stimando il RTT (Round Trip Time).

Vediamo un semplice procedimento che ci permette di impostare il valore del timeout di TCP. Poniamo SampleRTT come il tempo misurato intercorso tra la trasmissione di un segmento e la ricezione del suo ACK di riscontro: essendo però questo un valore variabile in funzione dello stato della rete, calcoliamo EstimatedRTT come una sua media mobile esponenziale ponderata, ovvero

$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}$$

dove un valore tipico per α è 0,125: in questa relazione l'influenza dei vecchi campioni decresce esponenzialmente in modo che siano più "pesanti" i valori più recenti, e quindi più reali, di RTT.

La successiva figura riporta il confronto tra i valori di RTT letti e quello del valore medio calcolato.



Generalmente il valore EstimtedRTT della media viene modificato aggiungendo un ulteriore margine di sicurezza ottenuto dal calcolo della deviazione standard DevRTT, così calcolata:

$$DevRTT = (1 - \beta) \cdot DevRTT + \beta \cdot |SampleRTT - EstimatedRTT|$$

L'intervallo di timeout è impostato come

$$TimeoutInterval = EstimatedRTT + 4 \cdot DevRTT$$

> Esistono alcune varianti dei procedimenti di calcolo del timeout: i più utilizzati si basano sugli algoritmi di Karn e Jacobson.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Con TPDU si intende:
a) Transport Data Protocol User
b) Transport Device Protocol User
c) Transport Device Protocol Unit
d) Transport Data Protocol Unit

2 La dimensione del payload TCP in ethernet è di:
a) 1500 byte
b) 1520 byte
c) 65495 byte
d) 65535 byte

3 Con MSS si intende:
a) minimun segment size
b) medium segment size
c) maximum segment size
d) nessuna delle precedenti

4 Quale tra i seguenti non è un FLAG del segmento TCP?
a) URG
b) ACK
c) RST
d) PSH
e) PST
f) SYN
g) FIN

5 Per aprire e chiudere la trasmissione vengono utilizzati i seguenti bit:
a) PST, SYN, FIN
b) RST, SYN, FIN
c) URG, SYN, FIN
d) ACK, SYN, FIN

Esercizi di completamento

1 Completa la seguente tabella, associando i servizi alle porte elencate 21, 23, 25, 69, 79, 80, 110, 119 e completando la tabella con la relativa applicazione.

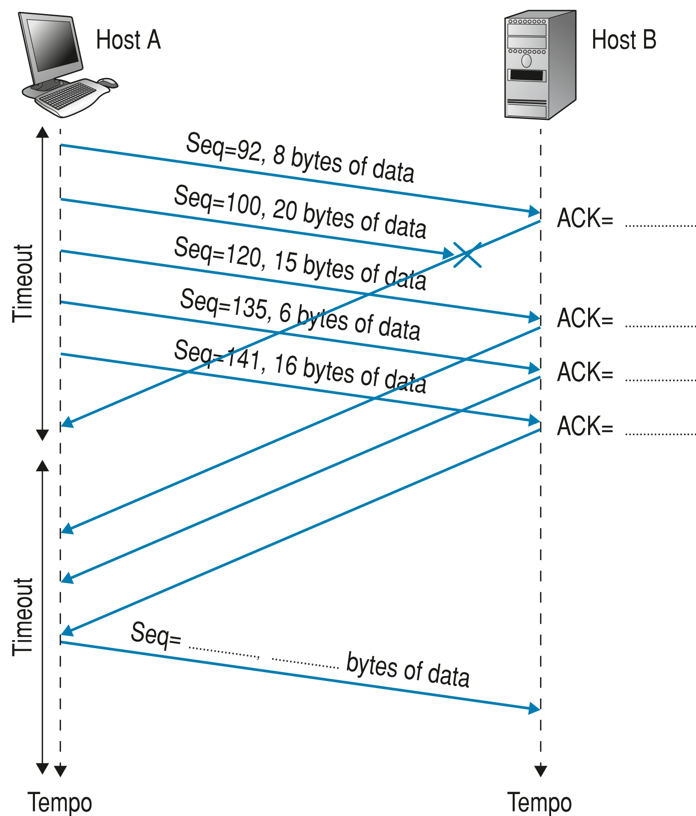
Porta	Protocollo	Applicazione
	POP-3	
	Telnet	
	Finger	
	SMTP	
	TFTP	
	NNTP	
	FTP	
	HTTP	

Test vero/falso

1 La caratteristica fondamentale del protocollo TCP è quella di essere un protocollo punto-punto. V F
2 Le porte con numero inferiore a 1024 non sono utilizzate nel protocollo TCP. V F
3 inetd permette l'esecuzione di un processo in grado di ridurre il carico del sistema. V F
4 Il valore di MTU è sempre superiore a quello del MSS. V F
5 Per sincronizzare il numero di sequenza iniziale il bit SYN deve essere settato a 1. V F
6 Ethernet impone come valore di MTU 1500 byte. V F
7 Il rilascio di una connessione è simmetrico e richiede la contemporaneità tra i due host. V F
8 Il valore ottimale del timeout è un parametro standard e dipende dal numero di hop. V F

Verifichiamo le competenze

1 **Indica tutti i valori di ACK nella situazione riportata nella seguente figura.**



Lezione 5

TCP: PROBLEMATICHE DI CONNESSIONE E CONGESTIONE

IN QUESTA LEZIONE IMPAREREMO...

- i problemi connessi con l'attivazione della connessione
- i problemi connessi con il rilascio della connessione
- la gestione della congestione della rete

Nella lezione precedente abbiamo descritto come avvengono le operazioni di attivazione e rilascio della connessione: queste due operazioni sembrano di semplice realizzazione ma in entrambe ci sono delle problematiche da risolvere che descriveremo in questa lezione.

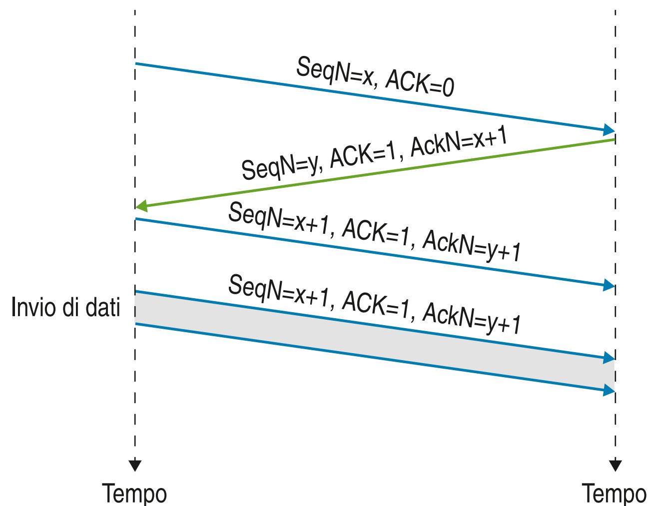
Problemi con l'attivazione della connessione

Durante la fase di attivazione della connessione, a causa di ritardi interni nell'invio degli ACK, la subnet può perdere o duplicare i pacchetti e può succedere che arrivino a destinazione due copie complete di pacchetti nella giusta sequenza e vengano quindi attivate contemporaneamente due connessioni per lo stesso canale.

La soluzione proposta nel 1975 da Tomlinson con il three-way handshaking risolve il problema dei duplicati relativi alla fase di attivazione della connessione.
Ricordiamo sinteticamente il funzionamento del protocollo:

1. il richiedente invia un TPDU di tipo conn.request con
   - A il flag SYN settato a 1,
   - B un numero x proposto come inizio della sequenza (l'Initial Sequence Number Seq=x),
   - C il flag ACK settato a 0;
2. il destinatario invia un TPDU di tipo ACK contenente:
   - A i flag SYN=1 e ACK=1,
   - B la conferma di x nell'Acknowledgment number (quindi invia AckN=x+1),
   - C il proprio numero iniziale di Seq=y (l'Initial Sequence Number);
3. il richiedente invia un TPDU di tipo dati contenente:
   - A il flag ACK=1,
   - B i primi dati del dialogo a partire da Seq=x+1,
   - C la conferma di y nell'Acknowledgment number (quindi invia AckN=y+1).

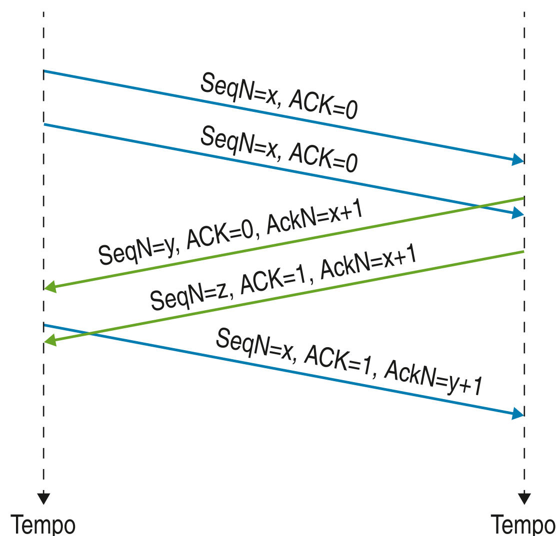
I valori x e y possono essere generati in modo da avere valori ogni volta diversi sfruttando, per esempio l'orologio di sistema.
Il funzionamento descritto è il seguente:



Analizziamo il comportamento in alcune possibili situazioni di malfunzionamento:

A Duplicazioni di richiesta di connessione

Nel caso di un errore, cioè se arriva a destinazione un duplicato della richiesta di attivazione, il destinatario risponde utilizzando un diverso numero di sequenza z e il mittente, che sa di non aver richiesto una seconda connessione, rifiuta tale risposta.



B Duplicazione di richiesta di connessione e di segmento

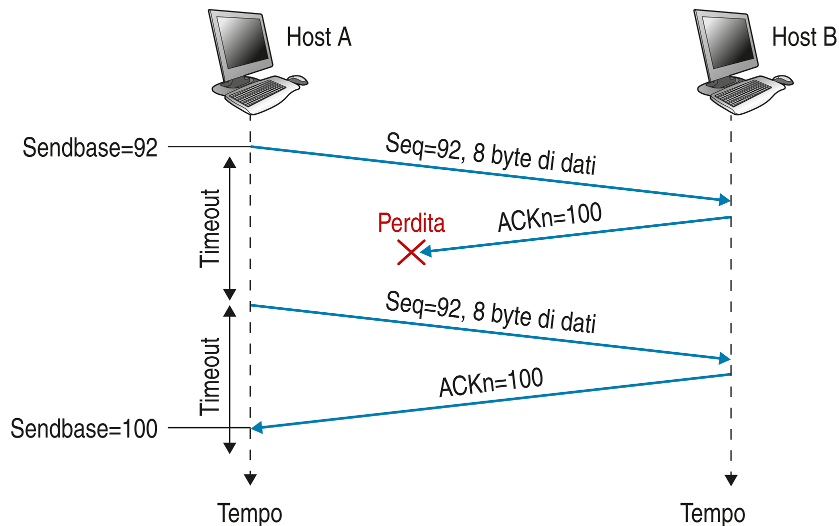
Anche nel caso in cui al destinatario arrivino sia un duplicato della richiesta di attivazione che un duplicato del primo TPDU dati non viene attivata la seconda connessione dato che:

- come nel caso precedente il mittente rifiuta la seconda risposta del destinatario perché sa di non aver richiesto una seconda connessione;
- il destinatario rifiuta il TPDU dati perché questo ha un ACK relativo a un numero di sequenza (**y**) precedente a quello (**z**) da lui inviato.

■ Problemi durante la connessione

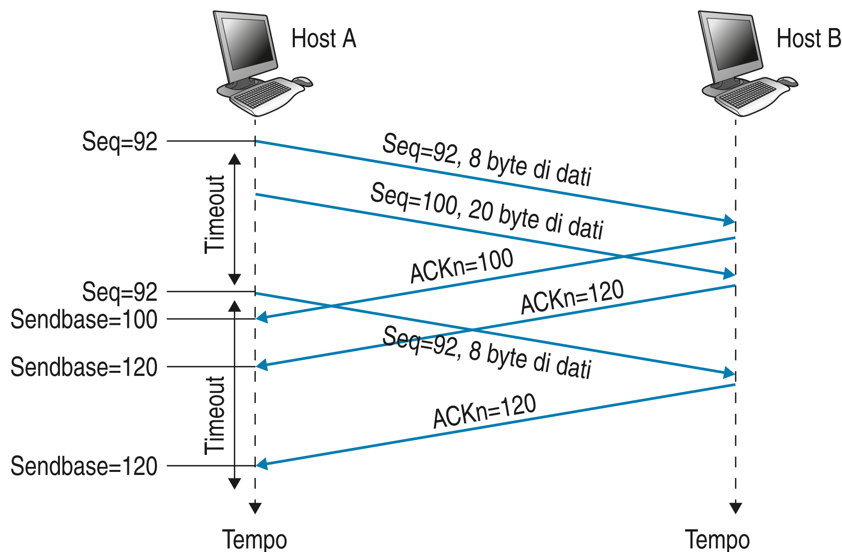
Vediamo le diverse situazioni nelle quali può trovarsi il mittente dopo la trasmissione del primo messaggio:

A nel caso in cui il mittente invii un solo messaggio nell'intervallo di tempo del timeout non può procedere con la successiva trasmissione se non gli perviene un ACK positivo di riscontro.
Se l'ACK di riscontro viene perso, ritrasmette il primo messaggio e solo all'arrivo del ACKn corrispondente può trasmettere la successiva sequenza e aggiornare il valore del Sendbase (nel caso riportato come esempio al valore 100).



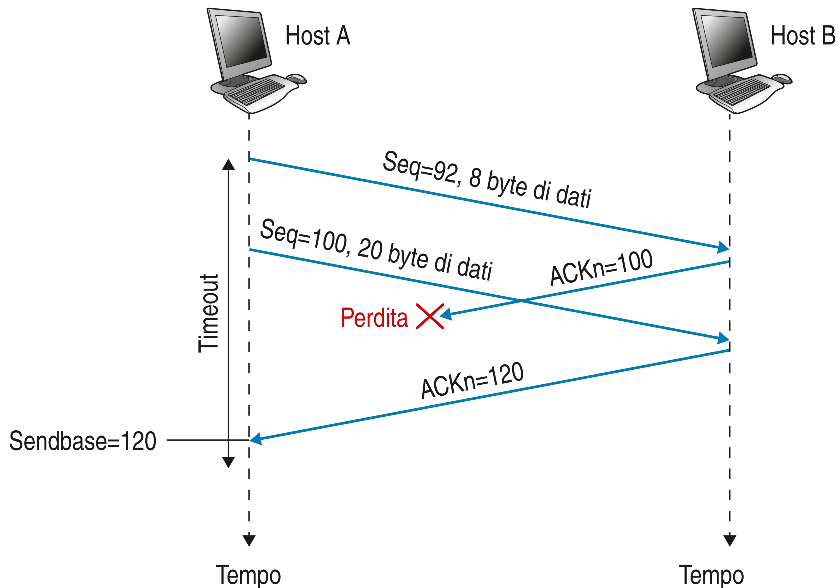
B Un secondo caso è quello che si verifica quando il mittente trasmette in sequenza due segmenti nello stesso timeout prima di attendere una conferma: se però il destinatario invia la (o le) conferma/e e queste non arrivano all'interno del timeout di attesa del mittente, questo ritrasmette il primo segmento (Seq=92) perché ipotizza la perdita di entrambe le trasmissioni.

Il ricevente riconosce che quest'ultima trasmissione è fuori sequenza e immediatamente gli invia un ACKn con il numero di sequenza atteso in modo che il mittente possa aggiornare il proprio Sendbase.



C Un terzo caso si verifica come il precedente quando il mittente trasmette in sequenza due segmenti nello stesso timeout prima di attendere una conferma: se viene perso un messaggio di risposta è sufficiente che al mittente giunga la conferma dell'ultima trasmissione, che ha il significato di conferma cumulativa di entrambe le trasmissioni.

Al suo arrivo, necessariamente però all'interno del timeout, aggiorna il Sendbase per le successive trasmissioni.



I messaggi di ACK

Analizziamo ora le possibili situazioni nelle quali il ricevente genera un messaggio di Acknowledgment:

A La situazione normale di comunicazione TCP è l'arrivo ordinato di un segmento con numero di sequenza atteso con tutti i dati già riscontrati fino al numero di sequenza corrente: il ricevente genera un ACK ritardato che può avere la durata fino a 500 ms per l'arrivo in attesa di un successivo segmento per non inserire traffico inutile sulla rete e confermare contemporaneamente entrambi i segmenti. Se invece nel timeout non arriva il segmento, invia una richiesta indicando l'opportuno ACKn atteso.

B Una seconda situazione nomale è quindi quella dell'arrivo ordinato di un successivo segmento prima della conferma del messaggio precedente: alla ricezione di questa seconda trasmissione il ricevente genera immediatamente un singolo ACK cumulativo con il numero ACKn totale, riscontrando in questo modo entrambi i segmenti ordinati.

C Se arriva un segmento con numero di sequenza inferiore a quello atteso, si tratta di un duplicato e quindi questo viene ignorato dal ricevente che genera immediatamente l'ACK con il numero ACKn finale, in modo che il mittente possa aggiornarsi.

D Se invece è in arrivo un segmento con numero di sequenza superiore a quello atteso viene a mancare la sequenza ordinata e si genera un buco: in questo caso il ricevente invia immediatamente un ACK indicando il numero di sequenza del primo byte mancante che possa colmare il buco.

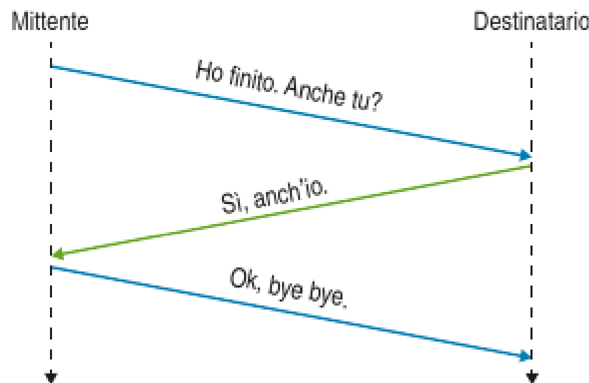
E L'ultimo caso analizzato è quello dell'arrivo di un segmento che colma parzialmente o completamente il buco: se il segmento comincia con il numero di byte uguale all'estremità inferiore del buco, il ricevente invia immediatamente un ACK di conferma indicante l'inizio del successivo elemento mancante, altrimenti rimane in attesa.

Problemi col rilascio di una connessione

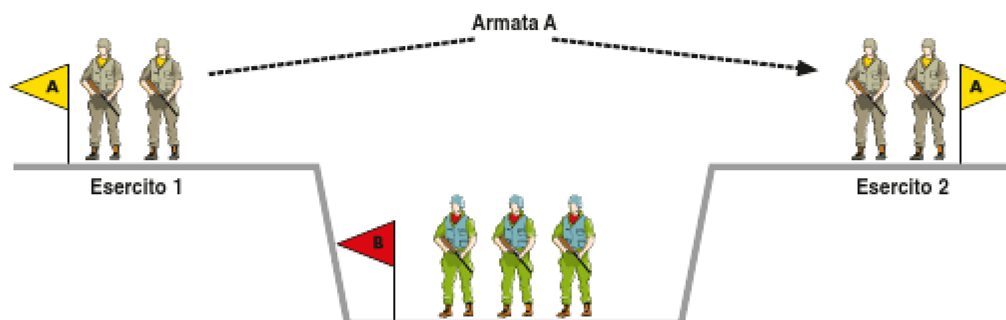
Anche il rilascio della connessione nasconde qualche insidia, nonostante sia più semplice della procedura che la realizza.
Con la procedura di rilascio ogni host deve rimuovere dalle proprie tabelle ogni informazione relativa alla connessione stessa e informare l'applicazione a livello superiore che la connessione non è più attiva.

Ci sono due tipi di rilascio, asimmetrico e simmetrico:
- nel primo caso uno dei due host chiude la connessione senza avvisare il secondo: esiste la possibilità di una perdita di dati, in quanto potrebbero essere presenti ancora sulla rete quelli che l'altra parte ha inviato e non sono ancora arrivati;
- nel secondo caso consideriamo di avere due connessioni unidirezionali che vengono chiuse ciascuna in modo autonomo, man mano che non ha più nulla da trasmettere: si potrebbe quindi avere un flusso di dati in una sola direzione quando l'altra è già chiusa.

Se invece si richiede che entrambi gli host chiudano contemporaneamente la connessione è necessario che "concordino" la chiusura, per esempio scambiandosi alcuni messaggi come i seguenti:



Sembrerebbe molto semplice sincronizzarsi per chiudere la connessione, ma il caso noto come il problema delle due armate ci smentisce.



La situazione è semplice: l'armata A è composta da due eserciti che singolarmente sono più deboli della armata B ma assieme hanno forze superiori: per poter quindi sferrare un attacco vincente è necessario che entrambi lo facciano contemporaneamente.
Per concordare l'inizio dell'attacco, però, i messaggeri devono passare nel territorio nemico e possono essere intercettati; come possono i due eserciti della armata A concordare l'ora dell'attacco?

Una soluzione potrebbe essere la seguente:
- il comandante dell'esercito 1 manda il messaggio "attacchiamo il giorno 22 a mezzanotte: attendo conferma";
- se il messaggio arriva a destinazione viene inviato un messaggio di conferma verso l'esercito 1; il comandante dell'esercito 2, però, non sa se questo messaggio è arrivato a destinazione, e quindi esita prima di effettuare l'attacco.

Per risolvere il problema proviamo a introdurre un ulteriore scambio di messaggi: quando arriva la risposta dell'esercito 2 all'esercito 1 si invia un ulteriore messaggio verso l'esercito 2 che "conferma la conferma".
Ma siamo sicuri che questo messaggio arrivi a destinazione? Ora il dubbio ce l'ha il comandante dell'esercito 1 e anche introducendo ulteriori scambi di messaggi non riusciremo mai a raggiungere la certezza, perché l'ultimo mittente avrà sempre un dubbio e quindi non esiste soluzione.

Se ora trasferiamo questa situazione nel rilascio del canale, possiamo osservare che il problema è identico e uno dei due host potrebbe non rilasciare mai la connessione.

Una possibile soluzione è quella di introdurre il Retransmission Time Out (RTT) già descritto: se non arriva una risposta entro un tempo prefissato, viene inviata nuovamente una richiesta per un massimo di n volte dopo di che si considera chiusa la connessione in ingresso e si provvede a chiudere anche la connessione in uscita.

Congestione di rete

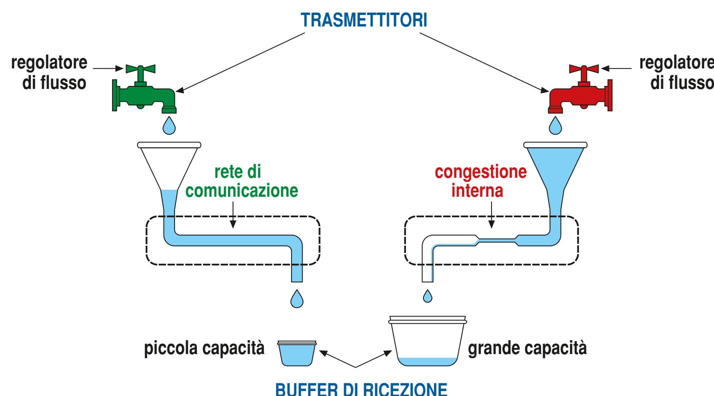
Gli stati di rete con i router e i canali trasmissivi sono utilizzati contemporaneamente da più applicazioni che creano più canali logici per scambiarsi tra di loro messaggi: la condivisione delle risorse può portare al fenomeno della congestione.



CONGESTIONE

Si può affermare che la rete Internet diventa congestionata quando il numero totale di messaggi che le applicazioni "mittenti" immettono nella rete nell'unità di tempo supera la velocità con cui i router e i canali trasmissivi della rete sono in grado di inoltrare i messaggi verso le applicazioni destinazione.

L'analogia rappresentata nel seguente disegno ci permette di descrivere semplicemente il problema.



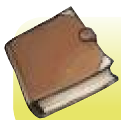
Se si verifica una congestione abbiamo come conseguenza due principali effetti negativi:
- rete bloccata parzialmente o completamente: il ritardo medio che ciascun datagramma accumula all'attraversamento di ciascun router cresce al crescere del livello di congestione e, al limite, diviene infinito;
- perdita di datagrammi: se un datagramma arriva a un router e trova il buffer completamente occupato (buffer overflow) viene scartato.

Ricordiamo il comportamento dei due protocolli dello strato di trasporto nel caso che un datagramma sia stato scartato a un certo router R per effetto della congestione:
- il protocollo UDP non fornisce un servizio di trasferimento dati affidabile e quindi il messaggio contenuto nel datagramma scartato non raggiungerà mai la destinazione: l'applicazione, quindi, avrà una perdita di informazione;
- il protocollo TCP fornisce un servizio di trasferimento dati affidabile e quindi il messaggio contenuto nel datagramma scartato verrà ri-trasmesso una o più volte da parte del processo-sorgente: dato che ogni ri-trasmissione richiede la ripetizione dell'introduzione in rete dello stesso messaggio, si incrementerà il traffico sulla rete che andrà a peggiorare il livello di congestione della rete.

> Il protocollo TCP considera le moderne linee di trasmissione molto affidabili e in caso gli ACK non tornino in tempo si attribuisce il problema alla congestione e non a un errore di trasmissione.

Il livello di trasporto si occupa della gestione della congestione perché è proprio lui che è in grado di regolare la frequenza e la velocità dei pacchetti presenti sulla rete: l'attività del TCP avviene non solo per rimediare alla congestione ma anche per *prevenirla oltre che curarla*.

La prevenzione viene attuata tenendo presenti due fattori:
- analisi della capacità della rete, tramite la definizione della finestra di congestione (CWND);
- individuazione del ricevitore lento, definendo la finestra di ricezione (RCVWND).



FINESTRA DI CONGESTIONE

Si definisce finestra di congestione o congestion window la finestra mantenuta dal mittente che rappresenta quanto si può spedire senza causare congestione.

Il dimensionamento della finestra di congestione varia in base agli eventi che il mittente osserva (ricezione ACK, timeout).
La dimensione della finestra di ricezione (o advertised window) dipende dalla disponibilità del buffer di ricezione che viene utilizzato per l'inoltro dei dati alle applicazioni.

Il rilevamento e controllo della congestione viene delegato al TCP ed essendo il TCP implementato solo negli host, il controllo di congestione è di tipo end-to-end: dato che TCP suppone che la perdita di pacchetti entro un certo tempo sia imputabile solo alla congestione, quando ciò avviene si interviene riducendo la velocità e quindi la frequenza delle trasmissioni.

Il modo più naturale per controllare il ritmo di immissione in rete dei dati per il TCP è quello di regolare la finestra di trasmissione: il trasmettitore non può trasmettere più del minimo tra RCVWND e CWND.
Cioè il valore massimo della finestra di trasmissione si ottiene da:

maxWindow = min (congestion window, advertised window)

Gli algoritmi che vengono utilizzati per il controllo della congestione sono descritti nella RCF2851: sono lo slow start, il congestion avoidance, il fast retrasmit e il fast recovery (avvio lento, evitare la congestione, ritrasmissione veloce e recupero veloce).

Questi algoritmi come idea di base intervengono sulla dimensione della finestra di congestione riducendola o aumentandola a seconda che il pacchetto giunga a buon fine o venga scartato. Vengono generalmente utilizzati in modo coordinato, anche se però non garantiscono che venga eliminata la congestione della rete.
Descriviamo sinteticamente i primi due algoritmi nel loro utilizzo coordinato che è stato proposto come versione di TCP chiamata TCP Berkeley, dal nome della università che l'ha perfezionata.

TCP Berkeley

La prima proposta di protocollo TCP che affrontava il controllo della congestione fu fatta a Berkeley nel 1986 introducendo gli algoritmi Slow Start e Congestion Avoidance e applicandoli congiuntamente.

In questa proposta il protocollo TCP non analizza la rete cercando di capirne lo stato ma agisce sul ritmo dell'invio dei dati quando questo raggiunge un particolare valore, la Slow Start Thershold (SSTHRESH), viene posta inizialmente a 64KB.

La finestra di congestione CWND viene determinata nel seguente modo: viene posta inizialmente pari alla dimensione del segmento massimo usato sulla connessione MSS e viene successivamente aggiornata con due diverse modalità:

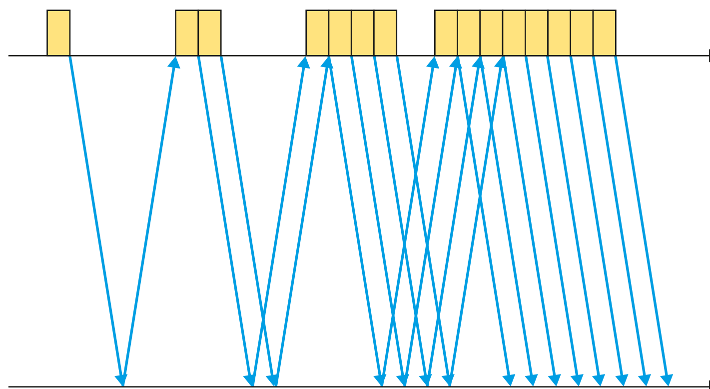
- se siamo nella situazione in cui CWND < SSTHRESH si utilizza l'algoritmo Slow Start;
- se siamo nella situazione in cui CWND > SSTHRESH si utilizza l'algoritmo Congestion Avoidance.

Descriviamo il funzionamento, a partire dall'accensione.

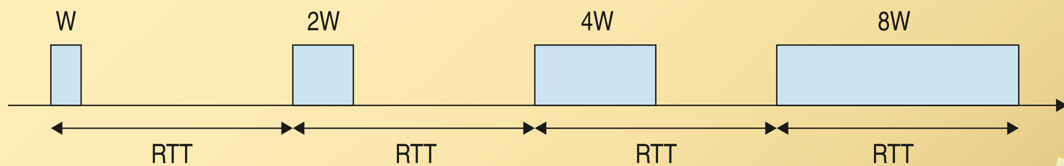
Slow Start

All'inizio il trasmettitore pone la CWND a 1 segmento MSS e la SSTHRESH a un valore di default molto elevato, generalmente 64KB: essendo CWND < SSTHRESH siamo nella situazione di Slow Start e si procede raddoppiando la CWND per ogni ACK ricevuto.

Si invia quindi un segmento e, se dopo RTT si riceve l'ACK, si pone CWND a 2 e si inviano 2 segmenti: se si ricevono 2 ACK, si pone CWND a 4 e si inviano 4 segmenti… e via di seguito fino a raggiungere il valore di SSTHRESH oppure fino a che si verifica una congestione.



Osserviamo che il nome Slow Start trae in inganno in quanto l'incremento della finestra avviene in modo esponenziale (raddoppia ogni RTT) che è quindi sicuramente non lento!



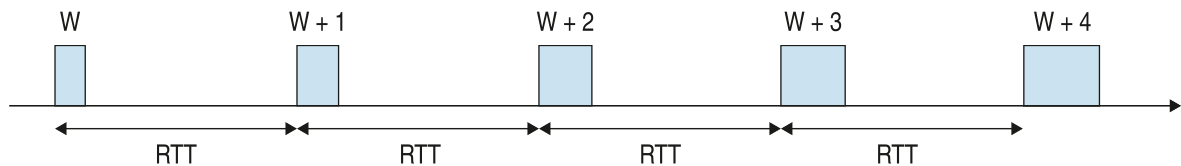
Con l'aumentare della finestra di trasmissione aumenta il rate di trasmissione che può essere stimato come:

$$R = \frac{CWND}{RTT} \text{ [bit/s]}$$

dove la CWND è espressa in bit e il RTT in secondi, sempre nella ipotesi che RCVWND > CWND.

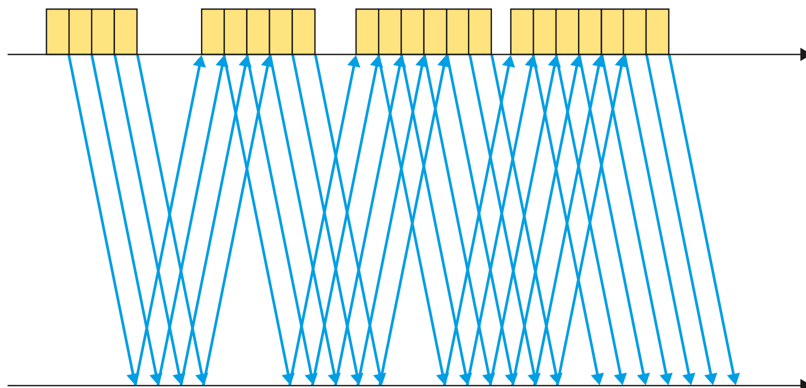
Congestion Avoidance

Abbiamo detto che lo **slow start** continua fino a che CWND supera la dimensione del SSTHRESH: oltre quel valore inizia la fase di Congestion Avoidance nella quale si incrementa la CWND di 1 MSS ogni RTT a ogni ACK ricevuto.



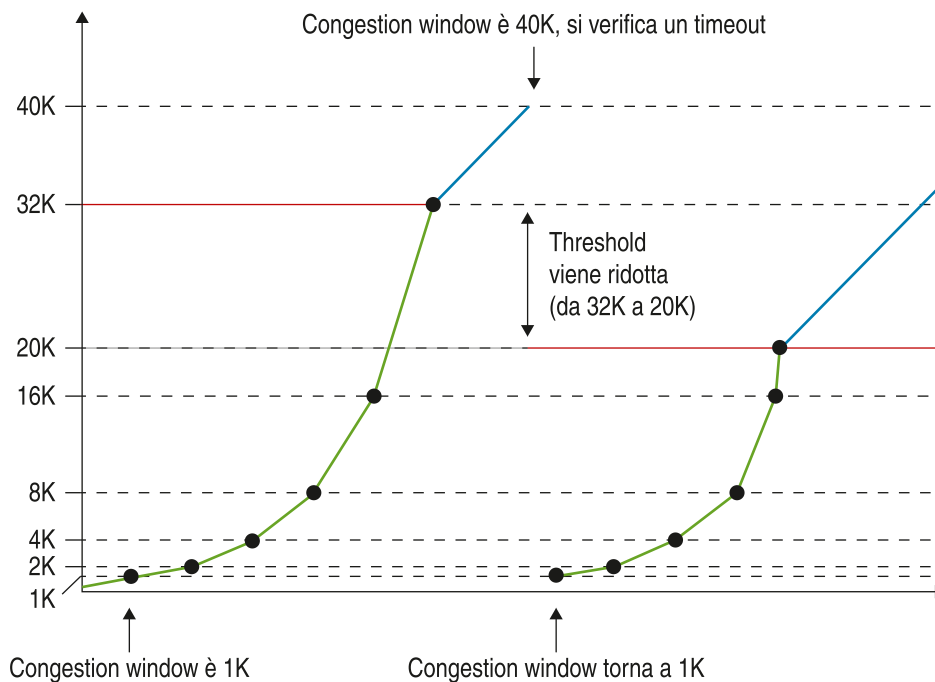
Si passa da un incremento esponenziale a un incremento lineare: dopo aver raggiunto SSTHRESH la finestra continua ad aumentare molto più lentamente che nella fase precedente.

In pratica se la CWND consente di trasmettere N segmenti, la ricezione degli ACK relativi a tutti gli N segmenti porta la CWND ad aumentare di 1 segmento.



Se si verifica la perdita di un segmento si ripete l'algoritmo da capo, dopo aver impostato il valore di threshold alla metà della dimensione della congestion window.

Nella seguente tabella riportiamo un esempio con segmenti di dimensione 1 Kbyte e prima threshold a 32 Kbyte.



Si può osservare la crescita esponenziale (in verde) fino al valore al quale è stata impostata la threshold, cioè 32 K, e la successiva crescita lineare (in blu) fino a che si verifica una congestione, per esempio quando raggiunge il valore di 40K.

Ora è necessario resettare l'algoritmo dopo però aver modificato il valore di threshold e averlo impostato alla metà del valore che precedentemente ha causato la congestione, quindi a 20K.
Quindi fino a 20K si utilizzerà lo Slow Start e successivamente il Congestion Avoidance, e via di seguito.

Verifichiamo le conoscenze

Domande a risposta aperta

1 Descrivi il protocollo three-way handshaking.

....
....
....
....
....
....
....

2 Descrivi sinteticamente i problemi connessi con l'attivazione della connessione.

a) ....: ....
....

b) ....: ....
....

3 Descrivi i problemi che possono verificarsi durante la connessione.

a) ....: ....
....

b) ....: ....
....

c) ....: ....
....

4 Descrivi sinteticamente le situazioni nelle quali viene generato un ACK.

a) ....
....

b) ....
....

c) ....
....

d) ....
....

e) ....
....

5 Descrivi le problematiche connesse con il rilascio simmetrico e asimmetrico.

a) ....
....

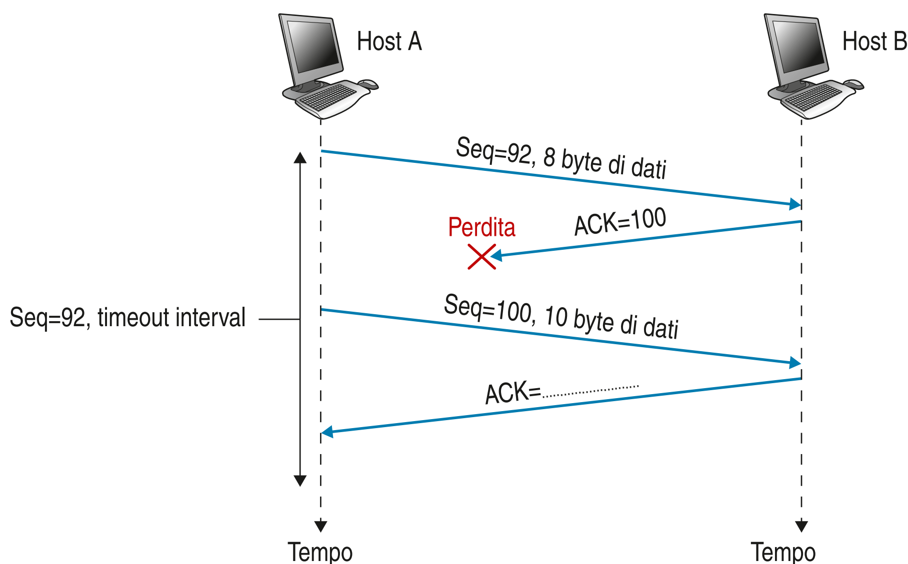
b) ....
....

Test vero/falso

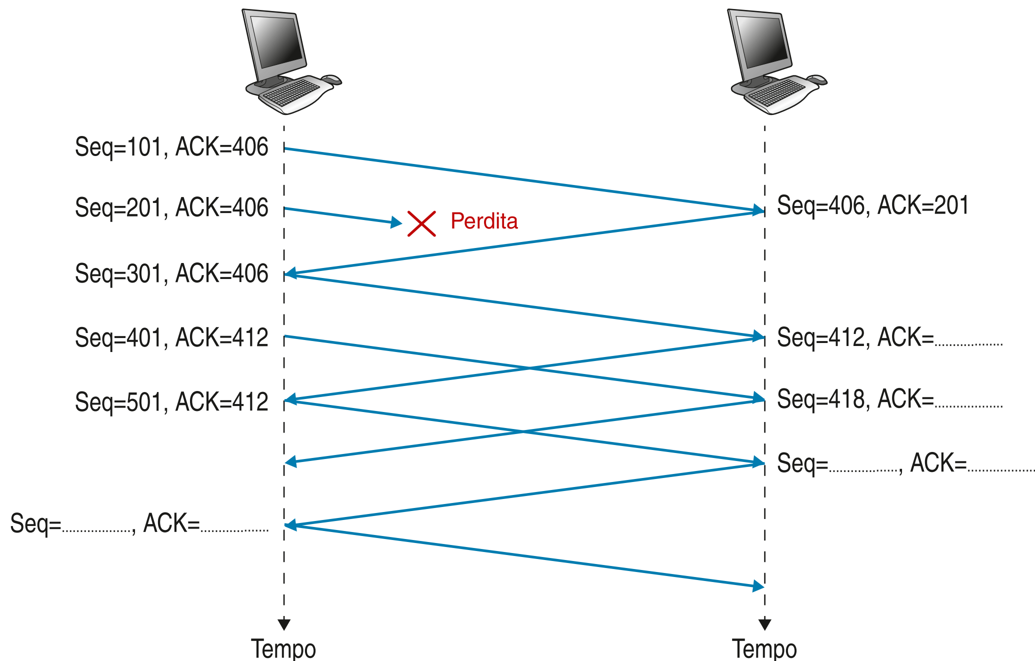
1 Nella fase di connessione si possono perdere pacchetti a causa di ritardi interni nell'invio degli ACK. V F
2 Il valore Sendbase viene aggiornato all'arrivo di un ACKn atteso. V F
3 Due segmenti inviati nello stesso timeout prima di attendere una conferma generano un errore. V F
4 TCP ammette la conferma cumulativa di più trasmissioni nello stesso timeout. V F
5 Nei messaggi di Acknowledgment, ACK deve sempre avere valore 1. V F
6 Nei messaggi di Acknowledgment, ACKn deve sempre avere valore 1. V F
7 Se nel timeout non arriva il segmento atteso si invia una richiesta indicando la Seq mancante. V F
8 Se arriva un segmento con numero di sequenza inferiore a quello atteso si genera un buco. V F
9 Il caso noto come il problema delle due armate nasce nel caso di rilascio asimmetrico. V F
10 Il Retransmission Time Out permette di risolvere i problemi connessi alla chiusura della connessione. V F
11 Il valore massimo della finestra di trasmissione si ottiene da: maxWindow = max (congestion window, advertised window). V F
12 Il TCP Berkeley applica congiuntamente il Fast Retrasmit e Congestion Avoidance. V F
13 Nel TCP Berkeley se CWND < SSTHRESH si utilizza l'algoritmo Congestion Avoidance. V F
14 Nel TCP Berkeley se CWND > SSTHRESH si utilizza l'algoritmo Congestion Avoidance. V F
15 Nel Congestion Avoidance si passa da un incremento esponenziale a un incremento lineare. V F

Verifichiamo le competenze

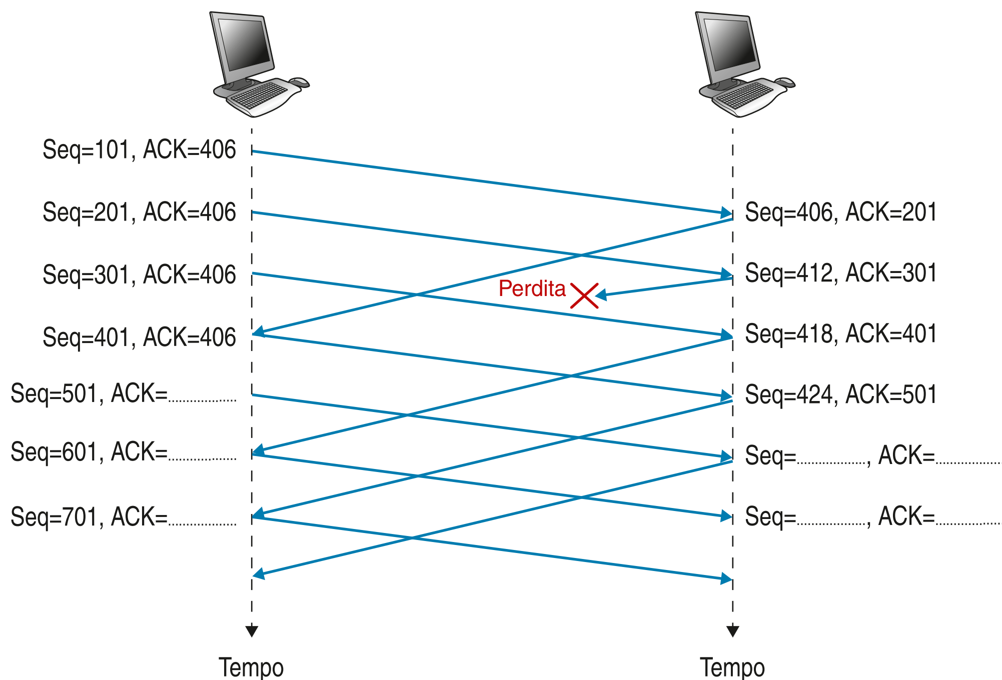
1 Qual è il valore di ACK che l'host B invia all'host A?



2 Completa il seguente diagramma temporale ipotizzando un errore nella trasmissione dei dati dove il client invia pacchetti di 100 byte e il server risposte di 6 byte e la window per entrambi è di 4 MSS.



3 Completa il seguente diagramma temporale ipotizzando un errore nella trasmissione dei dati dove il client invia pacchetti di 100 byte e il server risposte di 6 byte e la window per entrambi è di 4 MSS.



Esercitazioni di laboratorio 1

IL COMANDO NETSTAT

Il comando netstat permette di esaminare le connessioni attive e di visualizzare statistiche relative a connessioni di rete, tabelle di routing, interfacce di rete, masquerading e multicasting.

Vediamo le principali opzioni:

-i	visualizza una tabella di utilizzo delle interfacce di rete
-s	statistiche sommarie divise per protocollo
-v	output prolisso
-n	indirizzi numerici senza risoluzione dei nomi (più veloce)
-c	aggiorna la netstat ogni secondo
-p	visualizza PID e nome di ogni programma a cui ogni socket appartiene
-l	visualizza tutti i socket in modalità listening
-a	visualizza tutti i socket (listening e non listening)
-t	visualizza i dati relativi al protocollo TCP
-u	visualizza i dati relativi al protocollo UDP

netstat -n

Visualizza le informazioni relative alle connessioni TCP attive in un host.

```
C:\>netstat -n

Connessioni attive

  Proto  Indirizzo locale          Indirizzo esterno         Stato
  TCP    2.198.130.134:1370        70.84.167.203:3336        ESTABLISHED
  TCP    127.0.0.1:808             127.0.0.1:1044            ESTABLISHED
  TCP    127.0.0.1:1034            127.0.0.1:5820            ESTABLISHED
  TCP    127.0.0.1:1035            127.0.0.1:27015           ESTABLISHED
  TCP    127.0.0.1:1044            127.0.0.1:808             ESTABLISHED
  TCP    127.0.0.1:5820            127.0.0.1:1034            ESTABLISHED
  TCP    127.0.0.1:27015           127.0.0.1:1035            ESTABLISHED

C:\>_
```

Netstat permette quindi di monitorare l'attività delle porte del PC e quindi è uno strumento utile anche individuare eventuali aggressori che inviano richieste di syn.

netstat -s

Visualizza una serie di dati di tipo statistico sul traffico relativo ai diversi protocolli.

```
C:\WINDOWS\system32\CMD.exe

C:\>netstat -s

Statistiche IPv4

  Pacchetti ricevuti                    = 18882
  Errori di intestazione ricevuti       = 0
  Errori di indirizzo ricevuti          = 28
  Datagrammi inoltrati                  = 0
  Protocolli sconosciuti ricevuti       = 0
  Pacchetti ricevuti scartati           = 147
  Pacchetti ricevuti consegnati         = 18735
  Richieste di output                   = 17442
  Routing scartati                      = 0
  Pacchetti di output scartati          = 18
  Pacchetti output senza route          = 0
  Richieste di riassemblaggio           = 0
  Riassemblaggi riusciti                = 0
  Errori di riassemblaggio              = 0
  Datagrammi frammentati                = 0
  Errori frammentazione datagrammi      = 0
  Frammenti creati                      = 0

Statistiche ICMP

                            Ricevuti    Trasmessi
  Messaggi                  0           9
  Errori                    0           0
  Desti. irraggiungibile    0           9
  Tempo scaduto             0           0
  Problemi di parametro     0           0
  Quench sorgente           0           0
  Reindirizzamenti          0           0
  Echo                      0           0
  Risposte echo             0           0
  Timestamp                 0           0
  Risposte timestamp        0           0
  Mask indirizzo            0           0
  Risposte mask indirizzo   0           0

Statistiche TCP per IPv4

  Aperture attive                         = 503
  Aperture passive                        = 17
  Tentativi connessione non riusciti      = 9
  Connessioni reimpostate                 = 6
  Connessioni correnti                    = 11
  Segmenti ricevuti                       = 18341
  Segmenti trasmessi                      = 15723
  Segmenti ritrasmessi                    = 1114

Statistiche UDP per IPv4

  Datagrammi Ricevuti    = 385
  Nessuna porta          = 18
  Errori in ricezione    = 0
  Datagrammi trasmessi   = 551

C:\>
```



Prova adesso!

1. Visualizza tutti i socket riguardanti connessioni UDP.
2. Elenca i nomi dei processi server attualmente in esecuzione.
3. Avvia un server (per esempio apache) ed elenca i diversi stati in cui il processo si trova all'avvio.
4. Elenca tutte le connessioni di rete.
5. Elenca tutti i dispositivi attualmente connessi.

ESERCITAZIONI DI LABORATORIO 2

IL PROGRAMMA NMAP

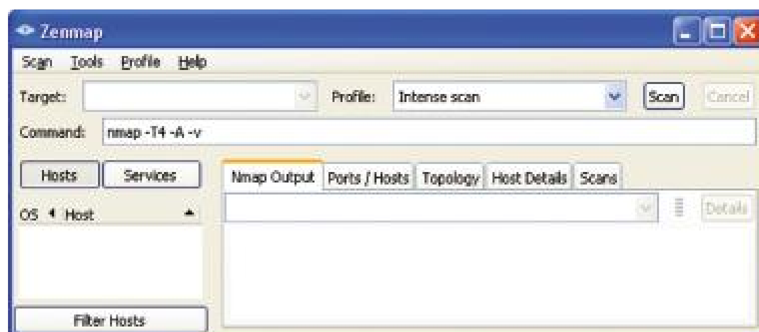
Nmap (Network Mapper) è un tool di rete open source per net exploration e auditing di sicurezza, progettato per controllare rapidamente reti anche di grosse dimensioni.

È un port scanner ampiamente utilizzato dagli esperti di sicurezza di rete (tra cui gli esperti di test di penetrazione e analisi forense): in questa esercitazione vedremo come effettuare le tecniche di scansione di base, le opzioni per l'host discovery, le opzioni di scansione delle porte, le tecniche utilizzate nella rilevazione del sistema operativo e dei servizi in esecuzione sul sistema.

Il programma è reperibile gratuitamente agli indirizzi http://nmap.org e http://www.insecure.org/nmap nella sezione download, oppure sul sito www.hoepliscuola.it, nella pagina riservata a questo volume, nella sezione materiali.



Dopo aver installato il pacchetto, è possibile sia digitare i comandi al prompt del sistema operativo oppure utilizzare la comoda interfaccia grafica riportata in figura:

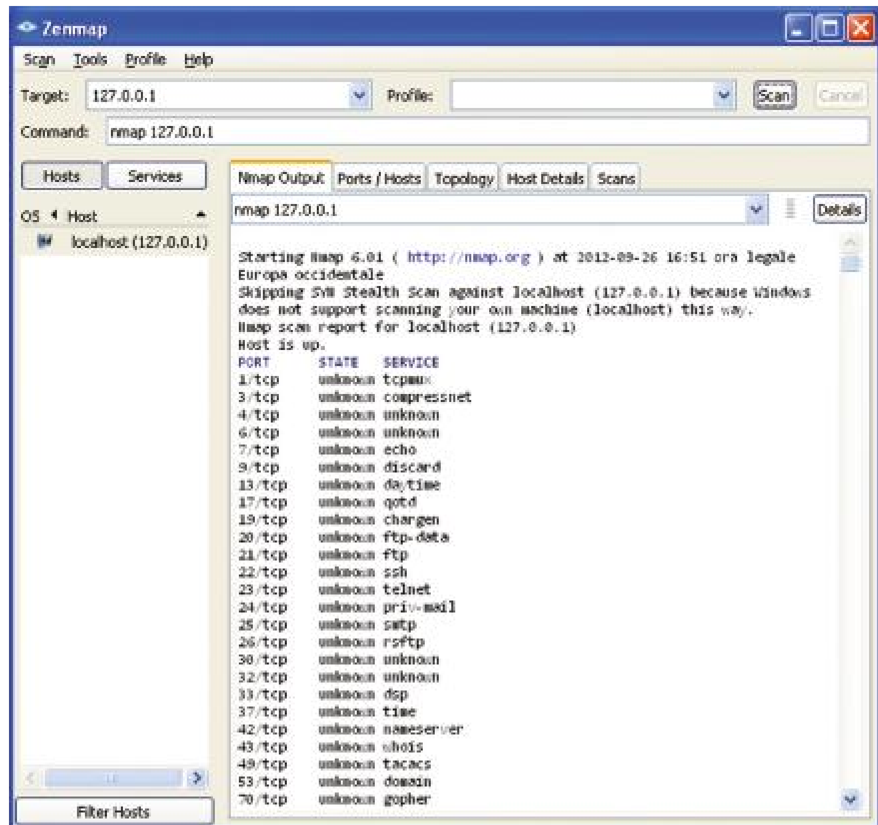


I comandi possono essere o selezionati da quelli predisposti nella tendina profile oppure digitati nella apposita riga di comando:

Command: nmap -sT

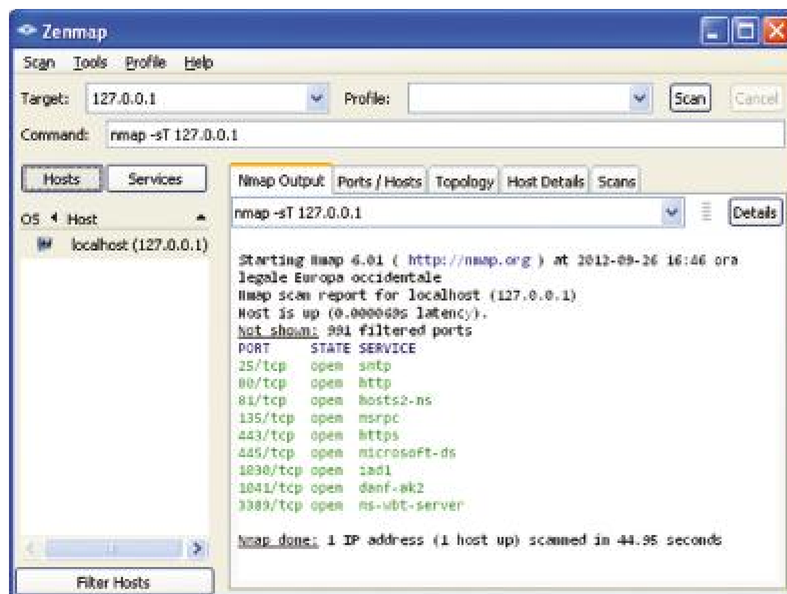
> Attenzione alla scrittura dei comandi con caratteri maiuscoli e minuscoli: è case sensitive.

Eseguiamo una prima scansione senza parametri, per esempio come la seguente:



quindi filtriamo le porte con il comando -sT che ci permette di individuare solo quelle aperte:

sullo schermo vengono elencate le porte con il loro numero e il tipo di protocollo, lo stato e il servizio offerto su quella porta.



Se non viene indicato alcun indirizzo IP viene effettuata l'analisi dell'host corrente: la sintassi completa del comando è la seguente:

```
scan -[tipo di scansione] -[opzioni] <host>
```

dove come host può essere scritto sia l'indirizzo IP che il nome simbolico dell'host.

Il tipo di scansione indica al programma la modalità con la quale deve effettuare l'analisi delle porte ed è una delle seguenti:

- -sT: avviene con modalità TCP connect, cioè controlla le porte tramite connect(): il risultato sarà l'elenco delle porte che la accettano (open) e di quelle che la rifiutano (closed);
- -sS: opzione che esegue il SYN scan, conosciuta come "strada mezza aperta" o stealth scanning , dove si inizia una connessione senza realmente terminarla (si invia un RST). Questo genererà una risposta corrispondente a un SYN/ACK per le porte aperte, un RST per quelle chiuse e nessuna risposta per quelle filtrate ("filtered");



◀ **Stealth Scan** Mechanism to perform reconnaissance on a network while remaining undetected. Uses SYN scan, FIN scan, or other techniques to prevent logging of a scan. ▶

- -sU: esegue la scansione sulle porte UDP aperte è particolarmente efficace sui sistemi Windows. Questa scansione è in grado di rivelare servizi (DNS, DHSP, SNMP) e trojans in esecuzione sulla macchina, tipicamente esposti a traffico UDP;
- -sP: questa scansione è chiamata "Ping Scan" e controlla tutti gli host di un range specificato che rispondono a un ping, permettendo di stabilire quali dispositivi sono online;
- -sO: questa scansione è chiamata "IP protocol scan" e determina i protocolli IP supportati dal target;
- -sI: l'opzione "idle scanning" è una tecnica avanzata di tipo stealth, che permette di scansionare un target senza inviare pacchetti che contengono traccia dell'origine;
- -sV: questa opzione, la "Version Detection", permette di raccogliere informazioni sui servizi in esecuzione in una macchina e in base a questi determinare la versione e il tipo di OS in esecuzione.

Le opzioni possono essere combinate tra loro e permettono di modificare la quantità delle informazioni visualizzate:

- -A: effettua la visualizzazione più completa disponibile;
- -d (debug): aggiunge all'output informazioni di debug;
- -oN fileName: salva l'output, in forma leggibile, nel file indicato;
- -F (Fast): semplifica e velocizza le operazioni di scansione analizzando solo le 100 porte più utilizzate;
- -O: fornisce indicazioni sul sistema operativo in esecuzione;
- -pNumero: esegue la scansione della sola porta indicata; è anche possibile eseguire la scansione per più porte con l'aggiunta di una virgola (,) tra ciascuna porta o eseguire la scansione di un intervallo di porte, è possibile indicarlo con un – (esempi: –p23 oppure –p 23,25,60 80-100).



Prova adesso!

Manda in esecuzione *nmap* utilizzando il qualificatore -sS e -O per l'individuazione del sistema operativo e l'indirizzo IP 127.0.0.1 come obiettivo.

Command: `nmap -sS -O 127.0.0.1`

Sappiamo che l'indirizzo IP 127.0.0.1 identifica il computer locale.

1. Quali porte aperte trova nmap?
2. Quali servizi e programmi stanno utilizzando queste porte?
3. Cerca di eseguire nmap con aperto un browser web o un client telnet. Questo cambia i risultati?

Manda ora in esecuzione nmap con la riga di comando precedente ma con l'indirizzo di un altro host presente sulla rete (per esempio, 192.168.1.7)

1. visualizza l'elenco dei socket in ascolto su una macchina target;
2. elenca i sistemi operativi (nome e versione) in utilizzo nella rete locale;
3. elenca delle macchine che hanno attivo un server di posta;
4. elenca dei servizi attivi sulla rete locale.

Ora esegui una "gara a coppie", cioè ciascuno cerca di controllare la macchina dell'altro alla scoperta del maggior numero di informazioni possibili e cercando di intercettare le tracce dell'intrusione del compagno sulla propria macchina.

Suggerimenti

- Host destinazione non risponde? Prova ad aggiungere l'opzione "-P0" che forzerà nmap per avviare la scansione, anche se si ritiene che la destinazione non esiste. Questa opzione è utile se il computer è bloccato da un firewall.
- Vuoi vedere a che punto è la scansione? Premi la barra spaziatrice, o qualsiasi tasto, mentre la scansione è in esecuzione, per visualizzare l'avanzamento di nmap.
- La scansione non termina dopo 5 minuti? Aggiungi l'opzione "-F" per la scansione nmap delle sole porte utilizzate più frequentemente.

Esercitazioni di laboratorio 3

LABORATORIO WIRESHARK: UDP

Premessa

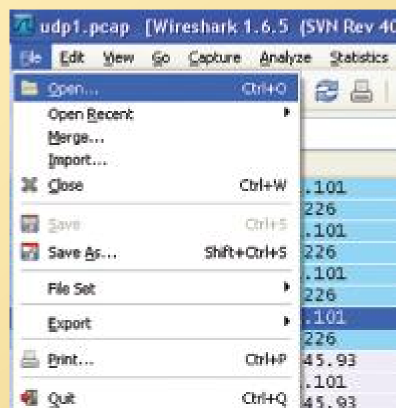
Wireshark è un programma in grado di osservare i messaggi scambiati tra entità di protocollo in esecuzione: rientra nella categoria dei packet sniffer in quanto, come suggerisce il nome, esso copia passivamente (ossia "sniffa, annusa") i messaggi che vengono inviati e ricevuti dal vostro computer; inoltre mostra i contenuti dei vari campi di protocollo e dei messaggi catturati.

La descrizione del funzionamento di Wireshark è stata fatta come esercitazione di laboratorio nel volume 1 del presente corso: è anche disponibile sul sito www.hoepliscuola.it nello spazio riservato al presente volume, nella cartella lezioni.
Il software è già incluso in tutte le versioni del sistema operativo Linux: è sufficiente effettuare l'installazione. Per gli altri sistemi operativi, è scaricabile gratuitamente dall'indirizzo http://www.wireshark.org/download.html oppure sempre dal sito www.hoepliscuola.it nello spazio riservato al presente volume, nella cartella materiali.

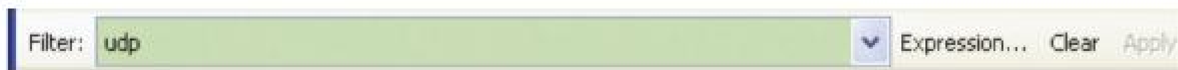
Analisi dei pacchetti UDP

In questa esercitazione useremo il packet sniffer Wireshark per analizzare i pacchetti UDP.
La prima operazione da eseguire è quella di intercettare dei pacchetti UDP inviati e ricevuti dal nostro elaboratore: per esempio, connettetevi al sito della rai http://www.rai.tv e mandate in esecuzione la ricezione di un programma in streaming.

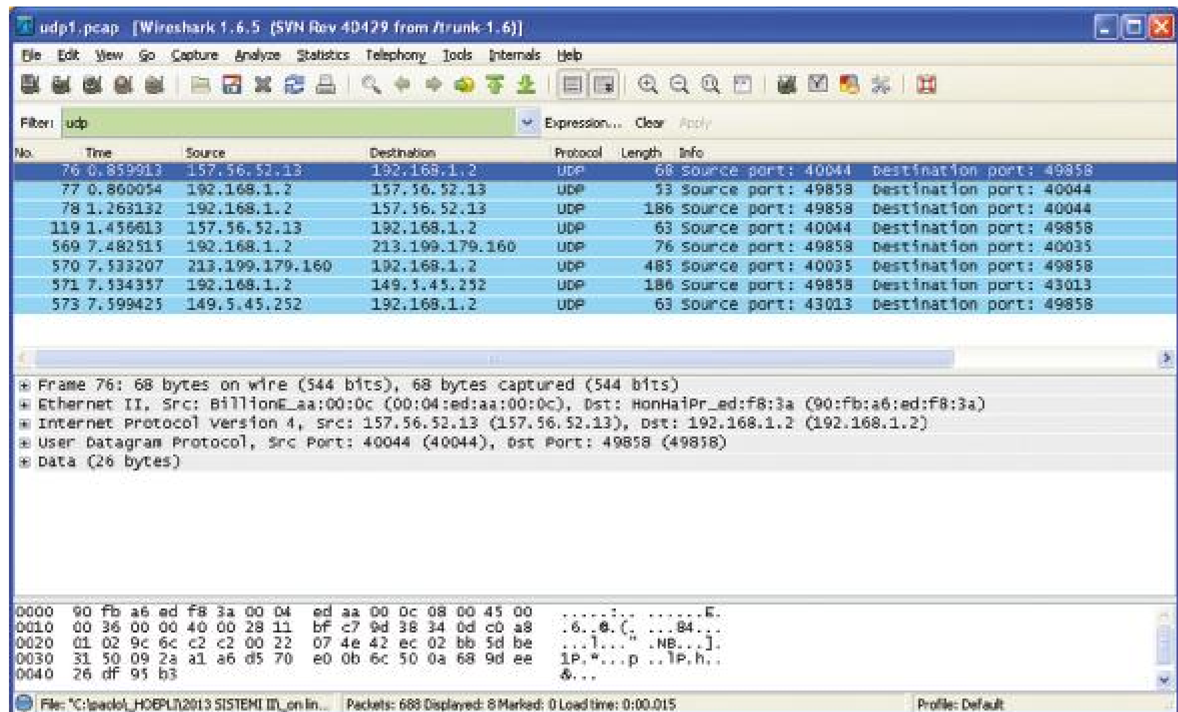
Se il laboratorio non vi permette di accedere direttamente a una connessione di rete, potete scaricare il file udp1.pcap dalla sezione source del sito riservato a questo volume dove sono presenti alcuni pacchetti catturati e caricarli all'interno di Wireshark usando il menù a discesa *File*, scegliendo *Open* e selezionando il file udp1.pcap.



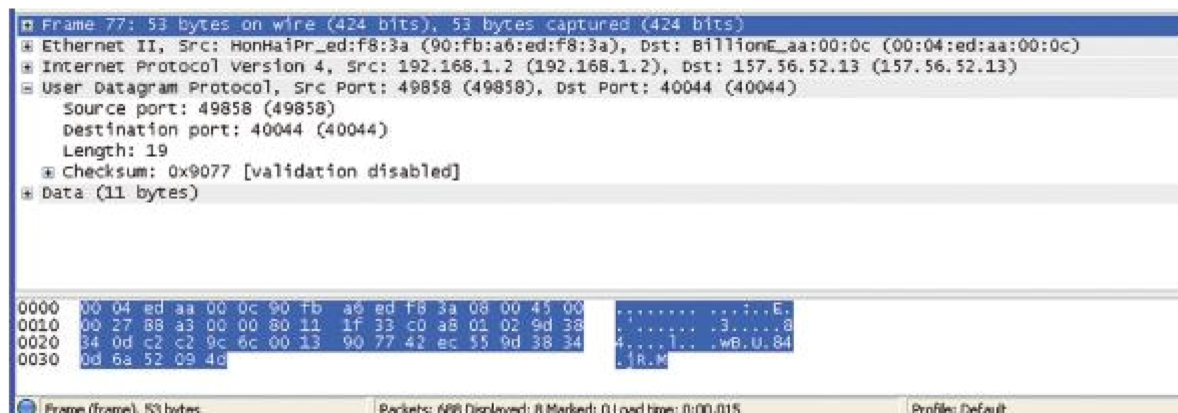
Dopo aver interrotto la cattura dei pacchetti, impostate il filtro in maniera che vengano visualizzati solo i pacchetti UDP inviati e ricevuti dal/al vostro sistema terminale.



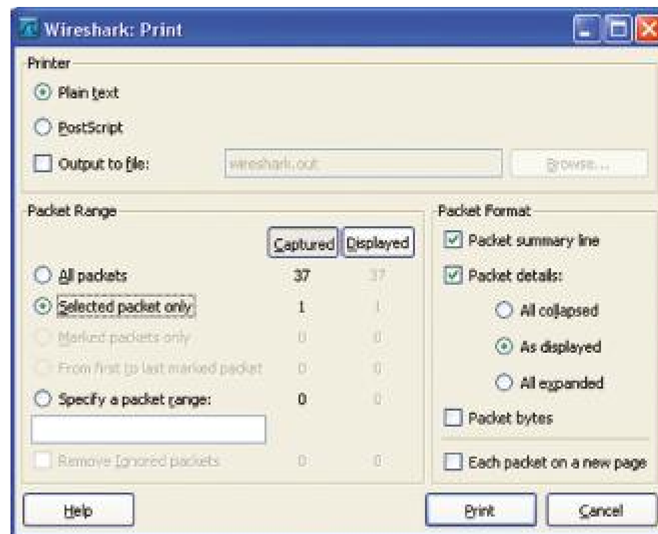
Il risultato è il seguente:



Prendete uno di questi pacchetti UDP ed espandete i campi UDP nella finestra dei dettagli.



Stampate il pacchetto usando File → Print, dopo aver scelto Selected packet only, scegliete Packet summary line e selezionate la minima quantità di informazioni sui pacchetti che è necessaria per rispondere alle domande.



Si consiglia di rispondere alle domande senza guardare il testo, per effettuare una autoverifica delle proprie conoscenze.



Prova adesso!

1. Seleziona un pacchetto e determina quanti campi ci sono in una intestazione UDP, evidenziandoli e indicando per ciascuno di essi il loro nome.
2. Analizzando la finestra inferiore dove è presente il contenuto del pacchetto, individua la dimensione in byte dell'intestazione UDP.
3. Dalla analisi della intestazione, individua il numero del protocollo UDP e riportalo sia come valore esadecimale che decimale.
4. Confronta i pacchetti catturati in modo da individuare a cosa si riferisce il campo "Length".
5. Quanti byte possono essere trasferiti con il protocollo UDP?
6. Qual è il valore più grande possibile per il numero di porta sorgente?
7. Esamina una coppia di pacchetti UDP che sono tra loro in relazione come richiesta-risposta: individua il numero di porta di entrambi e indica la relazione tra di esse.
8. Esegui manualmente il calcolo del checksum nel pacchetto verificando i risultati ottenuti e descrivendo ogni singola operazione effettuata.

Esercitazioni di laboratorio 4

LABORATORIO WIRESHARK: TCP

Esercizio 1: analisi dal download di un file

In questa unità di laboratorio esamineremo in dettaglio il funzionamento del protocollo TCP analizzando una traccia dei segmenti TCP inviati e ricevuti durante il trasferimento di un file di 150 KB (contenente il testo de l'*Inferno* di Dante Alighieri) dal vostro computer a un server remoto.

Inizieremo effettuando la cattura delle operazioni di download in modo da ottenere la traccia del trasferimento TCP di un file dal server remoto al vostro computer.

Il file inferno.txt è in formato ASCII ed è scaricabile dal sito http://www.hoepliscuola.it nella cartella materiali nella sezione riservata al presente volume.

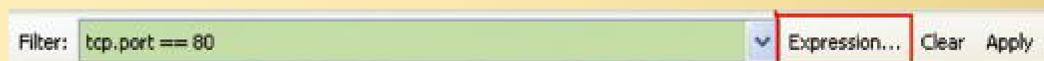
Esegui le seguenti operazioni per effettuare il download:

1. manda in esecuzione Wireshark;
2. avvia il tuo browser internet;
3. posizionati all'indirizzo http://www.hoepliscuola.it e scarica il file inferno.txt;
4. memorizza il file in una qualunque cartella del tuo calcolatore;
5. interrompi la cattura dei pacchetti.

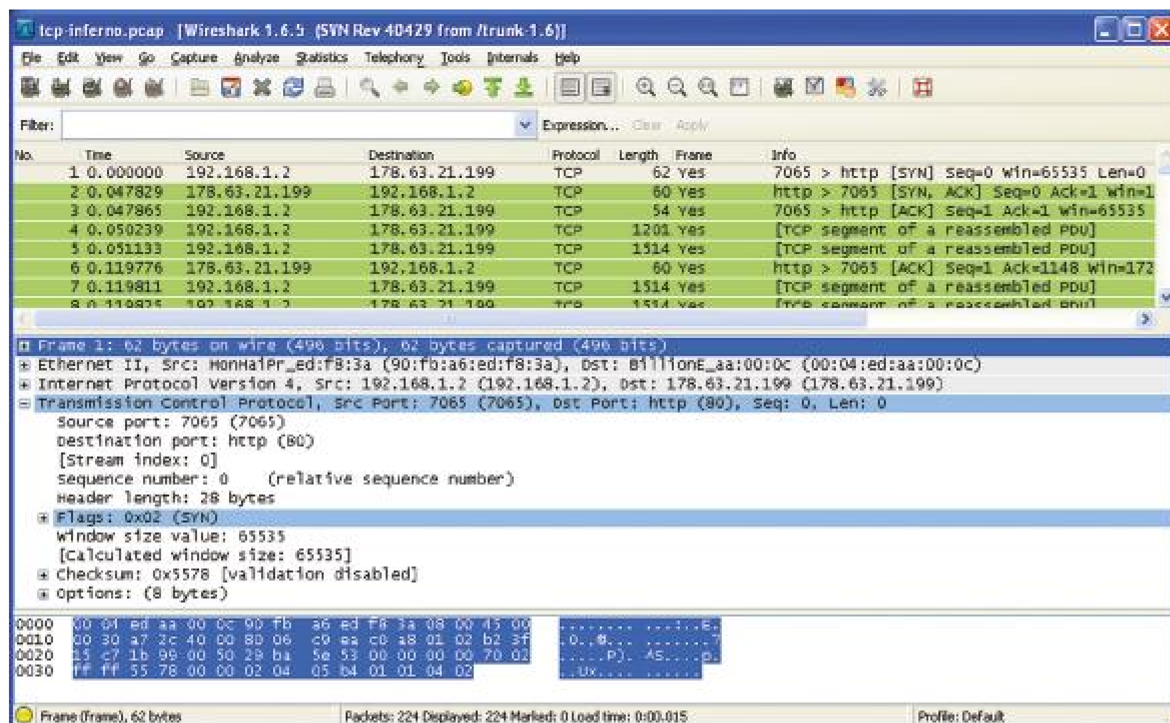
> Una copia del file contenente questi pacchetti è presente nel file TCP-inferno-DOWN.pcap scaricabile dal sito riservato a questo volume nella sezione materiali, utilizzabile nel caso non riusciste a effettuare voi le operazioni di cattura.

Per analizzare solo i pacchetti TCP è necessario impostare il filtro, che deve essere scritto immettendo `tcp` (in minuscolo) oppure `tcp.proto==80` in alto a sinistra, come in figura:

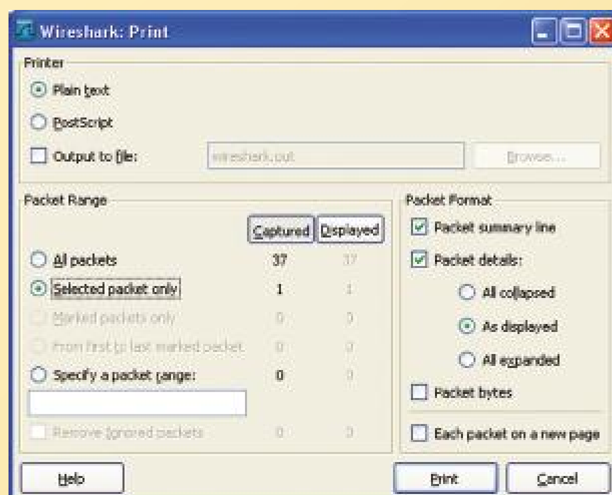
> Con `tcp.proto==80` si effettua un doppio filtro: dapprima si selezionano i pacchetti trasferiti col protocollo TCP e, in un secondo tempo, si selezionano quelli che hanno utilizzato la porta 80, cioè internet.
> Per inserire una espressione all'interno del campo filtro è possibile utilizzare la procedura guidata che si ottiene attraverso il bottone Expression.



Sullo schermo sono facilmente riconoscibili i pacchetti dell'*handshake a tre vie* iniziale contenenti un messaggio SYN ai quali fa seguito l'inizio del segmento HTTP con memorizzati nel payload i dati del file, leggibili in chiaro nella sezione di destra della finestra inferiore, dove i valori esadecimali sono anche visibili nel corrispondente carattere ASCII.



Per meglio rispondere alle domande si consiglia di stampare il pacchetto, in modo da poter evidenziare i campi e indicare direttamente sul diagramma le risposte.
La stampa si ottiene dal menu File con l'opzione *File* → *Print* con le opzioni indicate nella figura:



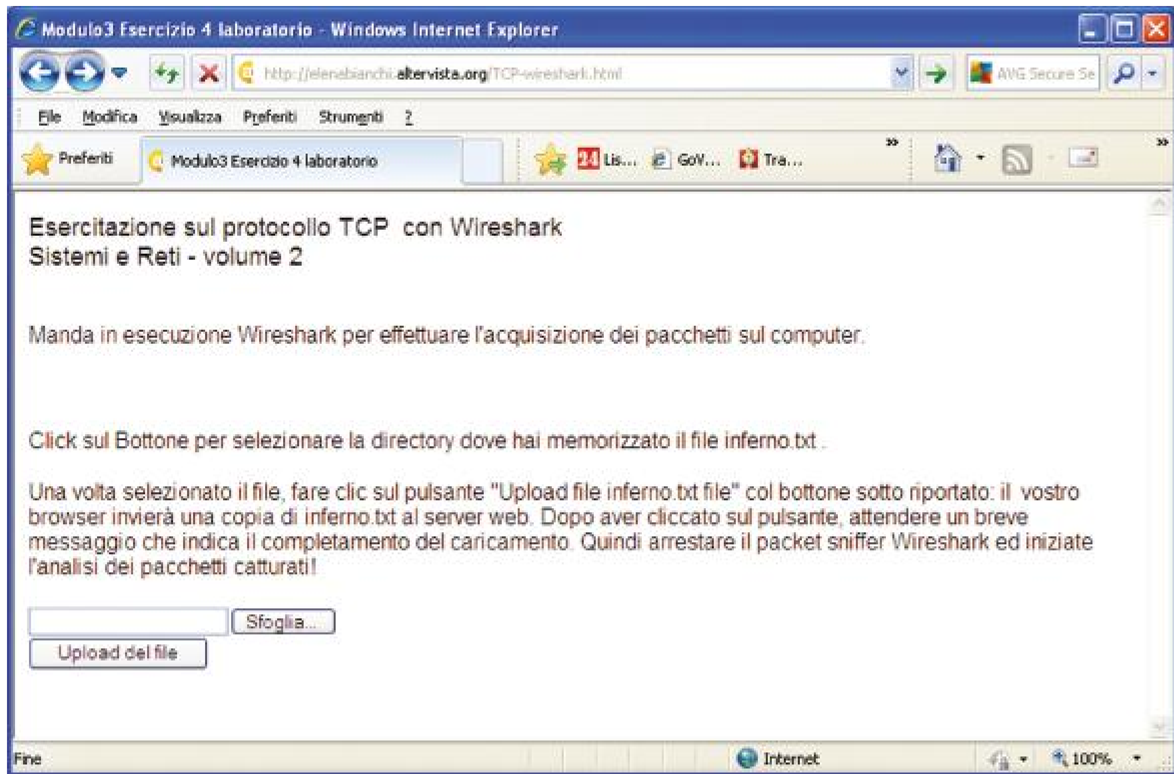


Prova adesso!

1. Individua l'indirizzo IP e la porta TCP usata dall'host sorgente che sta trasferendo il file.
2. Individua l'indirizzo IP e la porta TCP usata dall'host che sta ricevendo il file.
3. Evidenzia le tre fasi dell'handshake a tre vie
   - A) Quale elemento identifica un segmento come un segmento SYN?
   - B) Qual è il numero di sequenza del segmento TCP SYN che è utilizzato per inizializzare la connessione TCP tra i due computer?
   - C) Qual è il numero di sequenza del segmento SINACK inviato dal computer client come risposta al segmento SYN?
   - D) Cosa identifica questo segmento come un segmento SYNACK? Qual è il valore del campo ACK nel segmento SYNACK?
4. Analizza ora la sequenza di trasmissione dei dati del file
   - A) Come viene individuato il segmento HTTP POST?
   - B) Qual è il suo numero di sequenza TCP?
   - C) Individua i successivi tre segmenti: quali sono i loro numeri di sequenza e qual è la loro dimensione?
   - D) Individua per ciascuno il corrispondente messaggio di ACK e indica il valore dell'ACK number.
   - E) Qual è la lunghezza di ognuno dei primi sei segmenti TCP?
5. Utilizza ora la funzione che Wireshark mette a disposizione per disegnare l'RTT di ogni segmento TCP inviato, operando nel seguente modo:
   - A) seleziona un segmento TCP inviato dal client al server;
   - B) seleziona: Statistics → TCP Stream Graph → Round Trip Time Graph.
6. Analizzando le differenze tra messaggio trasmesso e ricevuto, indica il valore di RTT per i segmenti precedentemente analizzati.
7. Riesci a determinare la dimensione minima del buffer che viene indicata come disponibile in grado di ricevere l'intera traccia?
8. In questa trasmissione si è verificata la situazione di mancanza di spazio nel buffer ricevente che ha provocato un rallentamento della trasmissione?
9. Calcola infine il throughput per questa connessione descrivendo le singole operazioni.

Esercizio 2: analisi dall'upload di un file

Esegui le seguenti operazioni per effettuare l'upload del file inferno.txt, che viene realizzato utilizzando la seguente pagina raggiungibile all'indirizzo http://elenabianchi.altervista.org/TCP-wireshark.html



1. Posizionati sulla pagina sopra indicata;
2. utilizza il pulsante *Browse* presente in questa form per cercare il nome del file che hai salvato al primo punto. Non premere ancora il pulsante "Upload file inferno.txt";
3. manda in esecuzione Wireshark e inizia la cattura dei pacchetti;
4. ritorna al browser, e premi il pulsante "Upload file inferno.txt" per spedire il file al server;
5. al termine del caricamento del file viene visualizzato un breve messaggio di conferma nella finestra del browser;
6. interrompi la cattura dei pacchetti.

Rispondi alle medesime domande dell'esercizio precedente, confrontando pacchetto per pacchetto in modo da individuare le differenze tra download e upload.

> Una copia del file contenente questi pacchetti è presente nel file TCP-inferno-UP.pcap scaricabile dal sito riservato a questo volume nella sezione materiali, utilizzabile nel caso non riusciste a effettuare voi le operazioni di cattura.

4 LO STRATO DI APPLICAZIONE

UNITÀ DI APPRENDIMENTO

OBIETTIVI

- Conoscere il concetto di applicazione di rete
- Individuare le tipologie di applicazione di rete
- Avere il concetto di porta e di socket
- Conoscere l'architettura peer-to-peer (P2P)
- Comprendere il protocollo Telnet e il suo utilizzo
- Conoscere l'architettura gerarchica del WEB
- Comprendere i meccanismi del protocollo HTTP
- Acquisire il formato del messaggio HTTP
- Conoscere le funzioni del client e del server FTP

ATTIVITÀ

- Utilizzare le principali applicazione di rete
- Utilizzare un proxy server per navigare in modo anonimo
- Utilizzare i comandi Telnet
- Connettersi con una banca dati remota utilizzando Telnet
- Inviare una email utilizzando Telnet
- Acquisire le modalità di collegamento FTP
- Utilizzare i comandi FTP
- Analizzare i pacchetti HTTP con wireshark
- Analizzare i pacchetti SMTP e POP con wireshark

LEZIONE 1

IL LIVELLO DELLE APPLICAZIONI

IN QUESTA LEZIONE IMPAREREMO...

- il concetto di applicazione di rete
- le tipologie di applicazione
- le architetture client-server e P2P

Generalità

Nel modello ISO/OSI e TCP il livello delle applicazioni si occupa di implementare le applicazioni di rete che vengono utilizzate dall'utente finale. A tale livello, possiamo individuare Internet come la struttura più complessa esistente, in cui il programmatore non si deve preoccupare dei livelli inferiori ma soltanto utilizzare le primitive di comunicazione messe a disposizione dai diversi protocolli degli altri strati.

Ricordiamo che la pila protocollare di Internet è costituita da cinque livelli e l'unità di informazione che viene gestita a ogni livello assume forma e nome diverso in quanto a ogni passaggio vengono aggiunte le informazioni necessarie a ciascun protocollo per poter portare a termine i proprio compiti:

- al livello applicazione viene elaborato il tipo di dato messaggio;
- al livello trasporto viene elaborato il tipo di dato segmento;
- al livello rete viene elaborato il tipo di dato datagram;
- al livello collegamento (link) viene elaborato il tipo di dato frame;
- al livello fisico non corrisponde alcun tipo di dato (ma solo un segnale fisico).

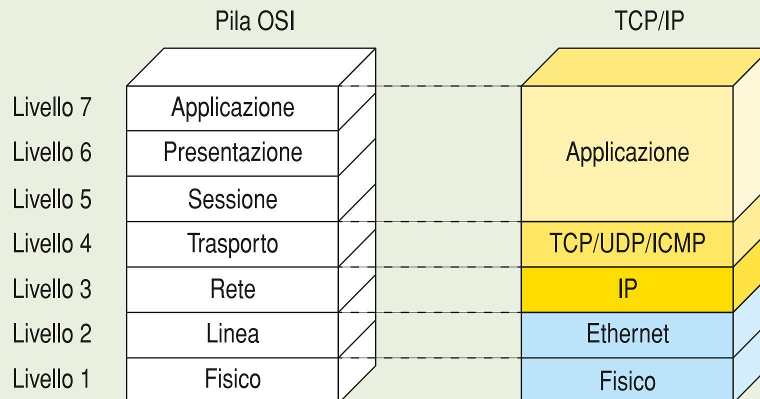
5 applicazione	messaggio
4 trasporto	segmento
3 rete	datagram
2 collegamento	frame
1 fisico	



Zoom su...

PILA ISO/OSI

Il modello a pila (stack) ISO/OSI può essere messo in relazione con altri stack protocollari, in modo da poterli comparare: la figura seguente mostra la pila TCP/IP, formata da 5 livelli, messa a confronto con la pila ISO/OSI:



Possiamo notare che la pila TCP/IP è analoga alla pila OSI, a eccezione del livello applicazione di Internet che racchiude i livelli 5, 6 e 7.

Il principale scopo delle reti, sia in locale che in remoto, è proprio quello di **condividere dati** mediante **applicazioni**.

Il livello applicazione implementa i vari protocolli, tra cui:
- SNMP: Simple Network Management Protocol;
- SMTP: Simple Mail Transfer Protocol;
- POP3: Post Office Protocol;
- FTP: File Transfer Protocol;
- HTTP: HyperText Transfer Protocol;
- DNS: Domain Name System;

che vengono utilizzati dalle seguenti applicazioni di rete:
- posta elettronica;
- web;
- condivisione di file P2P;
- giochi multiutente via rete;
- messaggistica istantanea;
- telefonia via Internet;
- videoconferenza in tempo reale;
- streaming di video-clip memorizzati;
- autenticazione in un calcolatore remoto (Telnet e SSH).

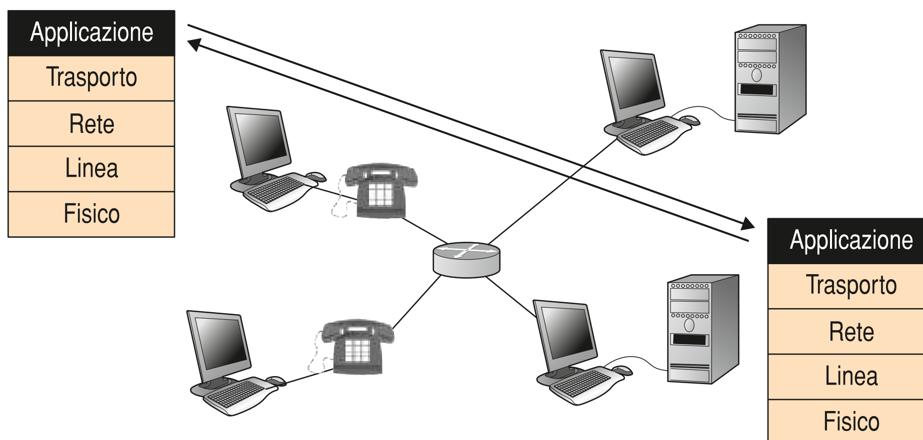
Oltre alle applicazioni generali offerte al pubblico di internet, sulla rete "gira" un infinito numero di applicazioni proprietarie, cioè sviluppate all'interno di una organizzazione per la gestione privata, come per esempio il collegamento tra filiali remote di una società commerciale, la connessione in tempo reale degli agenti col magazzino centrale ecc.

> Non va quindi confusa l'applicazione con il livello di applicazione: il livello di applicazione è lo strato protocollare che mette a disposizione i protocolli mediante i quali le applicazioni possono comunicare tra host remoti presenti sulla rete.

Applicazioni di rete

In generale una applicazione di rete è costituita da un insieme di programmi che vengono eseguiti su due o più computer contemporaneamente: questi operano interagendo tra loro utilizzando delle risorse comuni, accedendo cioè concorrentemente agli archivi (database), mediante la rete di comunicazione che li connette.

> L'applicazione di rete prende anche il nome di applicazione distribuita dato che non viene eseguita su di un solo elaboratore (concentrata).



I processi hanno la necessità di scambiare messaggi con gli altri processi della medesima applicazione, sia che essi appartengano alla stessa rete locale oppure che siano remoti e quindi dislocati dall'altra parte del globo: per comunicare tra loro questi processi devono mettersi in "contatto" tramite i loro indirizzi e utilizzando i servizi offerti dal livello di applicazione.



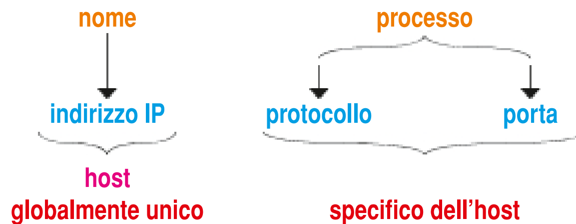
Zoom su...

API - APPLICATION PROGRAMMING INTERFACE

Le Application Programming Interface rappresentano un insieme di procedure, utilizzabili dal programmatore, utili alla stesura di applicazioni di rete. Le API forniscono l'interfaccia tra l'applicazione e lo strato di trasporto realizzando quella che si chiama astrazione tra hardware e programmatore, svincolando in tal modo gli sviluppatori dalle problematiche di comunicazione e trasferimento dati che sono i compiti degli strati inferiori.

Affinché un processo, presente su un determinato host, invii un messaggio a un qualsiasi altro host, il processo mittente deve identificare il processo destinatario in modo univoco. L'identificazione non può avvenire soltanto mediante l'indirizzo IP del destinatario, in quanto quest'ultimo individua soltanto l'host ma non il processo specifico. Pertanto l'identificazione deve tenere conto di due informazioni, l'indirizzo IP e il processo appartenente a quel determinato host, in sintesi:

- una identificazione del nodo su cui opera il processo con cui si desidera comunicare;
- una identificazione del particolare processo all'interno di quel nodo.



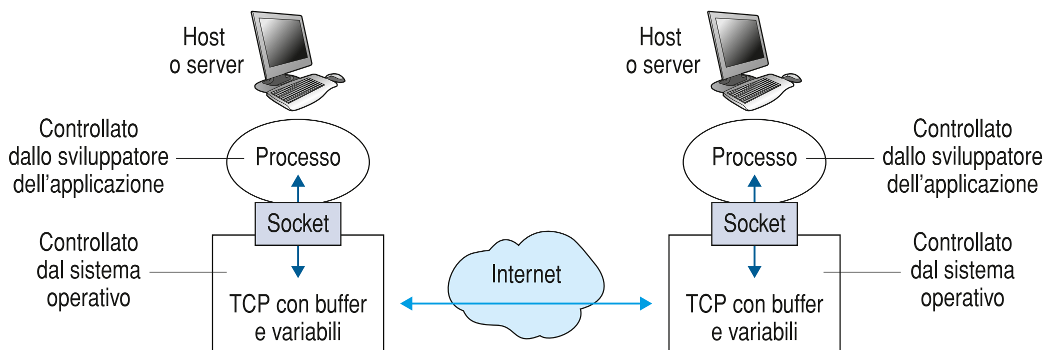
ESEMPIO 1 *Numeri di porte*

Un caso tipico è quello rappresentato da un server sul quale sono disponibili più servizi, tra i quali:

- email: viene inviata usando il protocollo applicativo SMTP, quindi occorre inviare un messaggio opportunamente codificato alla porta TCP 25 del server;
- sito web: per richiedere una pagina web si usa il protocollo applicativo HTTP, che invia un opportuno messaggio di richiesta alla porta TCP 80 del server;
- per trasformare un nome di un calcolatore in un indirizzo IP si invia una opportuna richiesta alla porta UTP 53 del server che offre il servizio DNS;

per richiedere il servizio a questo server è quindi necessario specificare l'indirizzo dell'host e il tipo di servizio desiderato.

L'identificazione univoca avviene conoscendo sia l'indirizzo IP che il numero di porta associato al processo in esecuzione su un host: questo meccanismo è già introdotto nelle lezioni precedenti e prende il nome di meccanismo dei ◀ socket ▶.

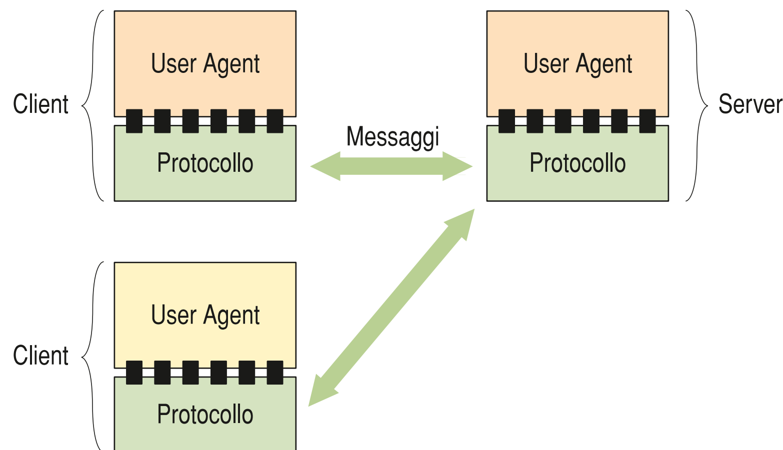




◀ Socket Come abbiamo visto nell'unità precedente un socket è formato dalla coppia <indirizzo IP:numero della porta>; si tratta di un identificatore analogo a una porta, cioè a un punto di accesso/uscita: un processo che vuole inviare un messaggio lo fa uscire dalla propria "interfaccia" (socket del mittente) sapendo che un'infrastruttura esterna lo trasporterà attraverso la rete fino alla "interfaccia" del processo di destinazione (socket del destinatario). ▶

Un socket consente quindi di comunicare attraverso la rete utilizzando la pila TCP/IP ed è quindi parte integrante del protocollo: le API mettono a disposizione del programmatore gli strumenti necessari a codificare la connessione e l'utilizzo del protocollo di comunicazione.

L'applicazione di rete può essere vista come composta da due parti:
- una user agent, che funge da interfaccia tra l'utilizzatore dell'applicazione e gli aspetti comunicativi;
- l'implementazione dei protocolli che permettono all'applicazione di integrarsi con la rete.



ESEMPIO 2 ***Componenti di un browser Web***

Possiamo individuare queste due componenti per esempio in un browser Web:
- l'interfaccia utente che serve a visualizzare i documenti ricevuti e a permettere la loro navigazione e a richiedere nuovi documenti specificando la loro URL;
- il motore del browser che è la parte che si preoccupa di inviare le richieste ai vari server e di ricevere le risposte.

Vedremo nella prossima lezione in dettaglio l'applicazione World Wide Web (WWW), che è costituita da numerose componenti, come i browser, i web server, i proxy server ecc. e utilizza molte convenzioni per codificare i dati, come HTML, CSS, ma per realizzare la comunicazione tra i diversi componenti sfrutta il protocollo HTTP (HyperText Transfer Protocol), che è implementato nel livello di applicazione.

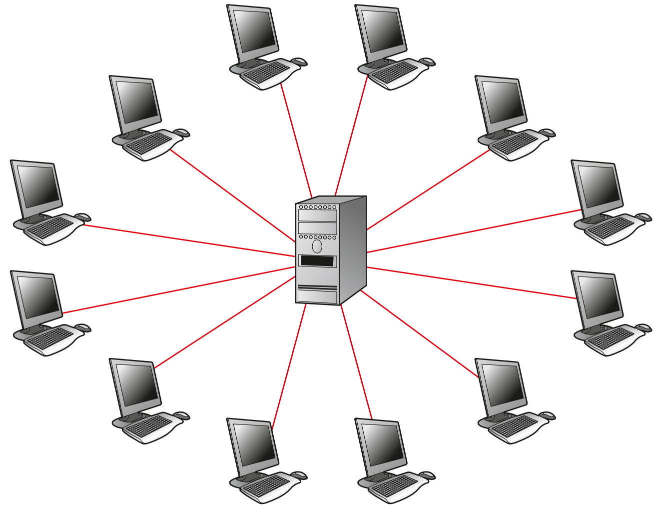
■ Architetture delle applicazioni di rete

Il primo passo che il programmatore deve effettuare per progettare una applicazione di rete è la scelta della architettura dell'applicazione; le principali architetture attualmente utilizzate sono le seguenti:
- client-server;
- peer-to-peer (P2P);
- architetture ibride (client-server e P2P).

Architettura client-server

Nella architettura client-server la caratteristica principale è che deve sempre esserci un server attivo che offre un servizio, restando in attesa che uno o più client si connettano a esso per poter rispondere alle richieste che gli vengono effettuate.

Un tipico esempio di questa architettura è il WWW, dove molteplici server (al limite uno per ogni sito pubblicato) possiedono le pagine (statiche o dinamiche) che saranno inviate ai client che ne fanno richiesta tramite i browser.
Il server deve sempre essere attivo e deve possedere un indirizzo IP fisso dove può essere raggiunto dagli host client: quindi l'indirizzo IP deve essere statico, contrariamente a quello dei client che generalmente è dinamico.
Un client non è in grado di comunicare con gli altri client ma solo con il server: più client possono invece comunicare contemporaneamente con lo stesso server.



Se un server viene consultato contemporaneamente da molti client potrebbe non essere in grado di soddisfare tutte le richieste e potrebbe entrare in stato di congestione: è necessario virtualizzare la risorsa realizzando una ◀ **server farm** ▶: questa non è altro che un server con un unico hostname ma con più indirizzi IP, trasparenti rispetto al client, sui quali vengono dirottate le richieste di connessione (viene utilizzato per esempio da Google, Amazon, Facebook e tutti i siti che hanno una elevata affluenza di visite).



◀ **Server farm** Server farm is a group of servers that are housed in one facility. A server farm comprises dozens, hundreds or even thousands of rack-mounted servers, typically running the same operating system and applications. Using load balancing, the workload is distributed among all machines. ▶

Architettura peer-to-peer (P2P)

Nelle architetture ◀ peer-to-peer ▶ (P2P) abbiamo coppie di host chiamati *peer* (persona di pari grado, coetaneo) che dialogano direttamente tra loro.
Nei sistemi P2P gli host possono essere visti come una comunità che collabora con il binomio dare e ricevere: ogni peer fornisce una risorsa e ottiene in cambio altre risorse.
Gli esempi più noti sono rappresentati dalle reti peer to peer in ambito di condivisione di file, come per esempio eMule, oppure Gnutella.

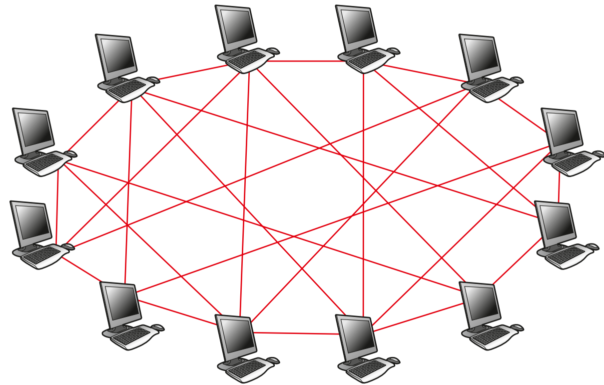
Un peer può anche decidere di offrire gratuitamente risorse, magari per la partecipazione a iniziative caritatevoli oppure di ricerca, come per esempio alla ricerca sul cancro oppure agli aiuti ai terremotati mediante donazioni.



◀ Peer-to-peer Un sistema Peer to Peer è formato da un insieme di entità autonome (peers), capaci di auto-organizzarsi, che condividono un insieme di risorse distribuite presenti all'interno di una rete. Il sistema utilizza tali risorse per fornire una determinata funzionalità in modo completamente o parzialmente decentralizzato. ▶

P2P decentralizzato

Nella architettura completamente decentralizzata un peer ha sia funzionalità di client che di server (funzionalità simmetrica = ◀ **servent** ▶), ed è impossibile localizzare una risorsa mediante un indirizzo IP statico: vengono effettuati nuovi meccanismi di indirizzamento, definiti a livello superiore rispetto al livello IP.
Le risorse che i peer condividono sono i dati, la memoria, la banda ecc.: il sistema P2P è capace di adattarsi a un continuo cambiamento dei nodi partecipanti (churn) mantenendo connettività e prestazioni accettabili senza richiedere l'intervento di alcuna entità centralizzata (come un server).

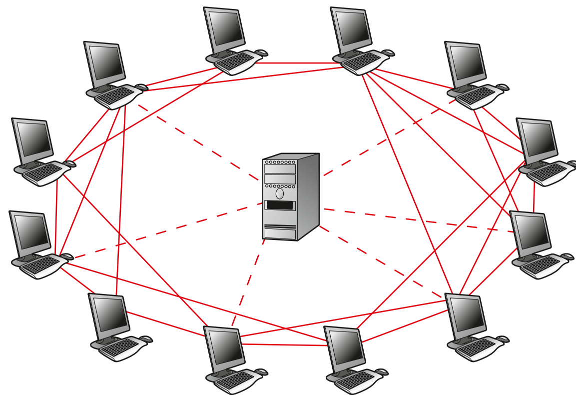




◀ **Servent** P2P knows of only one role, the servent, where the term "servent" is the combination of server and client ("SERVer + cliENT = SERVENT). Every servent performs, or at least is able to perform the same tasks (symmetric roles), which usually consist of searching its local or locally registered ressources, forwarding search requests and serving an offered ressource. ▶

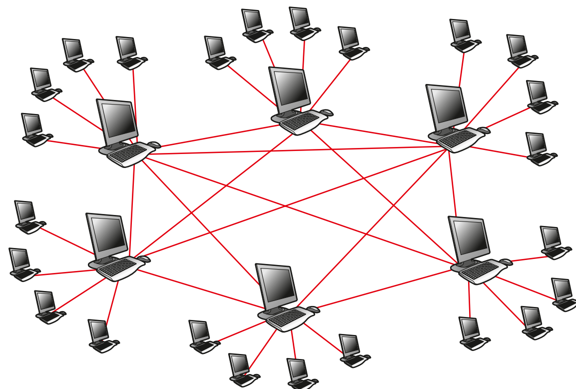
P2P centralizzato

Il P2P centralizzato è un compromesso tra il determinismo del modello client server e la scalabilità del sistema puro: ha un server centrale (directory server) che conserva informazioni sui peer (index, cioè il mapping resorse-peer) e risponde alle richieste su quelle informazioni effettuando quindi la ricerca in modalità centralizzata.
I peer sono responsabili di conservare i dati e le informazioni perché il server centrale non memorizza file, di informare il server del contenuto dei file che intendono condividere e di permettere ai peer che lo richiedono di scaricare le risorse condivise. La sua implementazione più famosa è Napster, dove gli utenti si connettono a un server centrale dove pubblicano i nomi delle risorse che condividono.



P2P ibrido (o parzialmente centralizzato)

Il P2P ibrido è un P2P parzialmente centralizzato dove sono presenti alcuni peer (detti supernodi o super-peer o ultra-peer) determinati dinamicamente (tramite un algoritmo di elezione) che hanno anche la funzione di indicizzazione: gli altri nodi sono anche chiamati leaf peer.





Zoom su...

APPLICAZIONI P2P

Tra le applicazione che utilizzano le architetture P2P ricordiamo le seguenti.

1. Condivisione di file (P2P file sharing):
   - Napster,
   - Gnutella, KaZaA,
   - eMule,
   - BitTorrent;
2. Telefonia e comunicazione (Instant Messaging and Voice over P2P):
   - IMP systems: Instant Message and Presence Applications,
   - VoP2P: Skype,
   - Jabber;
3. P2P TV:
   - Video Streaming applications;
4. Content Distribution Network (CDN):
   - CoralCDN;
5. P2P distributed storage:
   - Wuala,
   - Freenet;
6. Condivisione di risorse di calcolo:
   - SETI@home (Search for Extraterrestrial Intelligence).

Servizi offerti dallo strato di trasporto alle applicazioni

Le applicazioni richiedono allo strato di trasporto un insieme di servizi specifici oltre ai protocolli necessari per realizzarli, che possono essere standard o realizzati ad hoc.
Tutti i protocolli, sia standard che specifici, hanno in comune una particolarità: trasferire dei messaggi da un punto a un altro della rete. Ogni applicazione deve scegliere tra i protocolli di trasporto quale adottare per realizzare un protocollo applicativo in base ai servizi che sono necessari alle specifiche esigenze della applicazione, che possono essere riassunte in:

- trasferimento dati affidabile;
- ampiezza di banda;
- temporizzazione;
- sicurezza.

Trasferimento dati affidabile

Con trasferimento di dati affidabile intendiamo un servizio che garantisce la consegna completa e corretta dei dati: da una parte sappiamo che alcune applicazioni, come per esempio quelle di audio/video possono tollerare qualche perdita di dati senza compromettere lo scopo dell'applicazione mentre altre, come per esempio il trasferimento di file, richiedono un trasferimento dati affidabile al 100%.
A tale scopo il livello di trasporto mette a disposizione due protocolli:

- UDP User Datagram Protocol: il protocollo di trasporto senza connessione da utilizzarsi quando la perdita di dati è un fatto accettabile;
- TCP Transfer Control Protocol: il protocollo orientato alla connessione da utilizzarsi quando la perdita di dati è un evento inaccettabile, ovvero quando il trasferimento deve essere affidabile.

Ampiezza di banda (bandwidth) o throughput

Alcune applicazioni, come per esempio quelle multimediali, per poter "funzionare" hanno bisogno di avere una garanzia sulla larghezza di banda minima disponibile, cioè possono richiedere un throughput garantito di R bsp: si pensi alla trasmissione di un evento in diretta in una Web-TV.
Altre applicazioni, invece, non hanno questo bisogno come prioritario, ma utilizzano in modo elastico l'ampiezza di banda che si rende disponibile: tipici esempi sono la posta elettronica o i sistemi ftp.

Internet sappiamo avere una natura eterogenea e quindi i protocolli di trasporto non sono in grado di garantire la presenza di una certa quantità di banda per tutta la durata necessaria per trasmettere un messaggio ma, come vedremo, mette a disposizione la possibilità di implementare un protocollo applicativo flessibile che realizza un circuito virtuale che fallisce o si adatta se la banda è insufficiente.

Temporizzazione

Alcune applicazioni, come la telefonia Voip, i giochi interattivi, gli ambienti virtuali, per essere "realistiche" ammettono solo piccoli ritardi per essere efficaci: lo strato di trasporto non è in grado di garantire i tempi di risposta perché le temporizzazioni presenti assicurano un certo ritardo end-to-end tra le applicazioni.
Il protocollo TCP garantisce la consegna del pacchetto, ma non il tempo che ci impiega e neppure il protocollo UDP, nonostante sia più veloce del TCP, è temporalmente affidabile.

La soluzione, come vedremo, è quella di utilizzare un protocollo di trasporto in tempo reale, come RTP (Real Time Protocol), che è in grado di studiare i ritardi di rete e calibrare gli apparati e i collegamenti per garantire di restare nei limiti di tempo prefissati, scegliendo alternativamente quando utilizzare UDP e quando TCP.

Sicurezza

Una applicazione può richiedere allo strato di trasporto anche la cifratura di tutti i dati trasmessi in modo tale che anche se questi venissero intercettati da malintenzionati non si perda la riservatezza.
È quindi possibile che vengano richiesti dei servizi di sicurezza da applicare per garantire l'integrità dei dati e l'autenticazione end-to-end.

La tabella seguente riporta i requisiti richiesti al servizio di trasporto da parte di alcune applicazioni comuni.

Applicazioni	Tolleranza alla perdita dei dati	Ampiezza di banda	Sensibilità dal tempo
Trasferimento file	No	Variabile	No
Posta elettronica	No	Variabile	No
Documenti Web	No	Variabile	No
Audio/Video in tempo reale	Sì	Audio: da 5 Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Sì, centinaia di ms
Audio/Video memorizzati	Sì	Come sopra	Sì, pochi secondi
Giochi interattivi	Sì	Fino a pochi K	Sì, centinaia di ms
Messaggistica istantanea	No	Variabile	Sì e no

Conclusioni

Ricapitoliamo le caratteristiche dei due protocolli messi a disposizione dallo strato di trasporto per definire, in base alla specifiche richieste dalle categorie di applicazioni riportate nella precedente tabella, quale di essi è più opportuno utilizzare:

- UDP: il protocollo di trasporto senza connessione da utilizzarsi quando la perdita di dati è un fatto accettabile in quanto non è affidabile, non offre il controllo di flusso, il controllo della congestione, del ritardo e non garantisce una banda minima;
- TCP: il protocollo orientato alla connessione da utilizzarsi quando la perdita di dati è un evento inaccettabile, ovvero quando il trasferimento deve essere affidabile, dà la garanzia di un trasporto senza errori o perdita di informazioni, effettua il controllo di flusso in quanto se il ricevente è più lento del mittente esso rallenterà per non sommergere il ricevente, esegue anche il controllo della congestione limitando il mittente se la rete è sovraccarica, ma non dà garanzie di banda minima e ritardo minimo.

La seguente tabella riporta alcune applicazioni Internet dove è possibile confrontare il protocollo utilizzato a livello applicazione e il protocollo a livello di trasporto

Applicazioni	Protocollo a livello applicazione	Protocollo di trasporto sottostante
Posta elettronica	SMTP (RFC 2821)	TCP
Accesso a terminali remoti	Telnet (RFC 854)	TCP
Web	HTTP (RFC 2616)	TCP
Trasferimento file	FTP (RFC 959)	TCP
Multimedia in streaming	Proprietario (RealNetworks)	TCP o UDP
Telefonia Internet	Proprietario (Vonage, Dialpad)	Tipicamente UDP

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Quale tra i seguenti non è un protocollo applicativo?

a) HTTP
b) POP3
c) DNS
d) SMNP
e) SMTP
f) FTP

2 Quale tra le seguenti non è una applicazione di rete?

a) Posta elettronica
b) Condivisione di file P2P
c) Scheduler dei processi
d) Telefonia via Internet
e) Videoconferenza in tempo reale
f) TV in streaming
g) Telnet

3 Cosa significa SMTP?

a) Simple Message Transfer Protocol
b) System Mail Transfer Protocol
c) Simple Mail Transfer Protocol
d) System Message Transfer Protocol

4 Effettua il corretto accoppiamento:

1	.... messaggio	a)	datagramma
2	.... trasporto	b)	frame
3	.... rete	c)	segmento
4	.... collegamento	d)	applicazione

5 Quale è il significato di API?

a) Application Protocol Internet
b) Application Protocol Interface
c) Application Programming Interface
d) Application Programming Internet

6 Quali tra le seguenti non sono architetture di una applicazione di rete?

a) client-server
b) B2B
c) B2C
d) P2P

7 Quale tra i seguenti servizi offerti non è garantito dallo strato di trasporto alle applicazioni?

a) trasferimento dati affidabile
b) ampiezza di banda
c) velocità di comunicazione
d) temporizzazione
e) sicurezza

Test vero/falso

1 Il protocollo HTTP è a livello di trasporto. V F
2 Il protocollo TCP è a livello di applicazione. V F
3 L'applicazione di rete prende anche il nome di applicazione distribuita. V F
4 L'applicazione di rete prende anche il nome di applicazione multiutente. V F
5 Sullo stesso host possono essere in esecuzione molti processi. V F
6 Sullo stesso host può essere in esecuzione una sola applicazione. V F
7 Un protocollo è una parte integrante di una applicazione ed è sviluppato all'interno di essa. V F
8 In una architettura client-server gli indirizzi IP devono essere statici. V F
9 Il termine **servent** è ottenuto dalla contrazione **server-client**. V F
10 Nel P2P decentralizzato i servent hanno indirizzo IP statico. V F

Esercizi di completamento

1 Completa le seguenti tabelle.

Applicazione	Tolleranza alla perdita dei dati	Ampiezza di banda	Sensibilità dal tempo
Trasferimento file			
Posta elettronica			
Documenti web			
Audio/video in tempo reale			
Audio/video memorizzati			
Giochi interattivi			
Messaggistica istantanea			

Caratteristiche del protocollo	UDP	TCP
Orientato alla connessione		
Trasferimento affidabile		
Garantisce trasporto senza errori		
Effettua il controllo di flusso		
Esegue anche il controllo della congestione		
Dà garanzie di banda minima		
Dà garanzie di ritardo minimo		

Applicazione	Protocollo applicativo	Protocollo di trasporto sottostante
Posta elettronica		
Accesso a terminali remoti		
Web		
Trasferimento file		
Multimedia streaming		
Telefonia internet		

Domande aperte

1 Descrivi i servizi che devono essere offerti dallo strato di trasporto alle applicazioni di livello superiore.

2 Descrivi cosa sono e come funzionano i socket.

3 Quali sono le differenze tra le architetture delle applicazioni.

4 Quali protocolli di trasporto sceglieresti per realizzare un gioco di ruolo su rete? Perché?

5 Quali protocolli di trasporto sceglieresti per realizzare una simulazione di Gran Premio di Formula 1? Perché?

LEZIONE 2

IL PROTOCOLLO TELNET

IN QUESTA LEZIONE IMPAREREMO...

- il protocollo Telnet
- i comandi di Telnet
- gli utilizzi di Telnet

Generalità

L'utility ◀ **Telnet** ▶ (telecommunications network) è stata uno dei primi strumenti disponibili su Internet e permette agli utenti autorizzati di collegarsi ad altri computer ed eseguire i programmi presenti su di essi.
In sostanza fornisce all'utente la possibilità di effettuare un login remoto all'interno di una rete (remote login).
Al protocollo è stata assegnata la porta 23 e si basa su una connessione TCP per inviare dati e comandi in formato ASCII codificati a 8 bit: permette semplicemente di instaurare un sistema di comunicazione bidirezionale (half-duplex).

Il protocollo Telnet è definito nella RFC 854 e anch'esso è un protocollo del livello applicazione e fornisce le regole di base per permettere di collegare un client (costituito semplicemente da un monitor e una tastiera) a un interprete di comando (lato server).



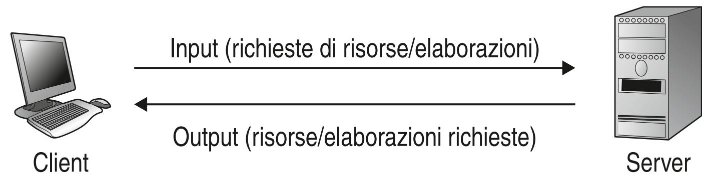
◀ **Telnet** "The purpose of the Telnet protocol is to provide a fairly general, bidirectional, eight-bit byte oriented communications facility. Its primary goal is to allow a standard method of interfacing terminal devices and terminal oriented processes to each other..." ▶

Nelle specifiche della prima versione di Telnet non è presente né l'autentificazione né la cifratura dei dati, che circolano in chiaro sulla rete: ora all'atto della connessione viene richiesto il nome utente e la password, ma, come vedremo, sono facilmente intercettabili dato che anch'essi viaggiano come i dati senza cifratura.

Il protocollo Telnet

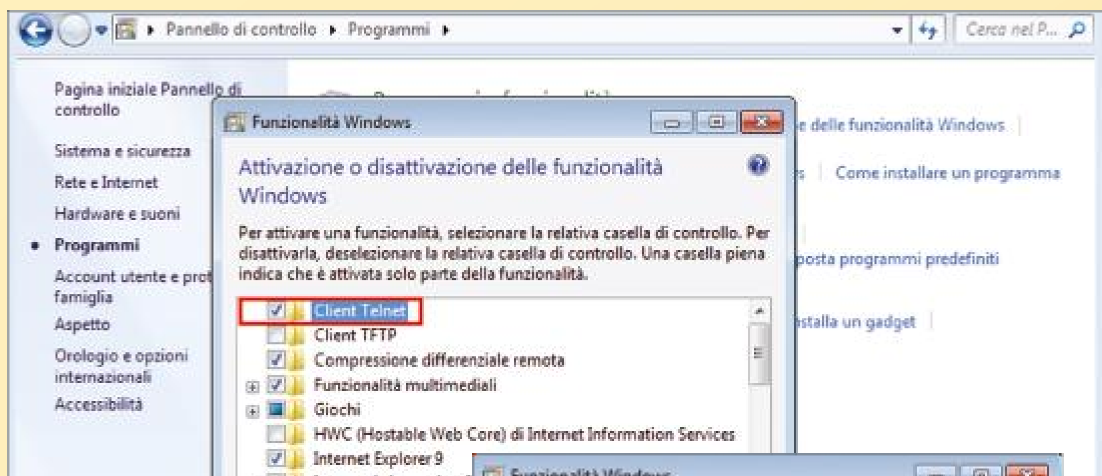
Il protocollo Telnet fu sviluppato a partire dal 1969 e fu uno tra i primi protocolli del livello applicativo: oggi il suo utilizzo è fortemente sconsigliato in reti pubbliche, per motivi legati alla sua mancanza di sicurezza.

Per utilizzare il protocollo Telnet occorre un programma client, di norma distribuito con il sistema operativo, che fornisce un'interfaccia a caratteri verso un interprete di comandi in esecuzione su un host remoto: il server.



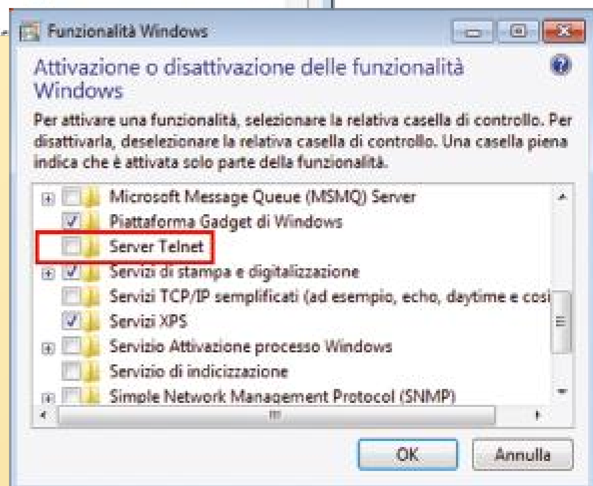
Le versioni odierne dei sistemi operativi non hanno abilitato di default tale client, ma è necessario attivarlo selezionando l'apposita voce dal pannello di controllo.
Per esempio, per Windows 7, la procedura è la seguente:

1. Fare clic sul pulsante *Start*, su *Pannello di controllo* e infine su *Programmi*.
2. In *Programmi e funzionalità* fare clic su *Attivazione* o *disattivazione* delle funzionalità Windows. Se viene chiesto di fornire una password amministratore o una conferma, digitare la password o confermare.
3. Nella finestra di dialogo *Funzionalità Windows* selezionare la casella di controllo *Client Telnet*.



4. Fare clic su OK. L'installazione potrebbe richiedere alcuni minuti.

In modo analogo, se si vuole rendere la propria macchina server Telnet, è necessario abilitare la funzionalità sempre nel pannello di controllo:



Per avviare una sessione Telnet occorre indicare l'host al quale collegarsi: di default la porta utilizzata è la porta 23, ma è anche possibile indicarne una diversa mediante la linea di comando:

```
telnet nomehost
```

oppure

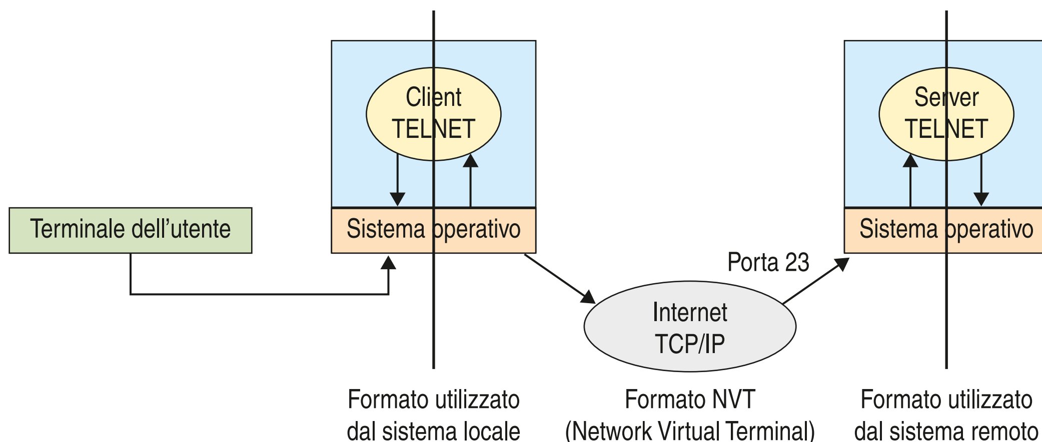
```
telnet nomehost 35
```

La comunicazione avviene mediante un protocollo bidirezionale half-duplex codificato a 8 bit e il flusso di dati viene inviato linea per linea: il byte che segue un valore 255 viene interpretato come un comando.

Il protocollo Telnet si basa su tre concetti fondamentali:
- il paradigma del terminale virtuale di rete (NVT, *Network Virtual Terminal*);
- il principio di opzioni negoziate;
- la simmetria del collegamento.

La nozione di terminale virtuale

Lo scopo del Terminale Virtuale di Rete o NVT è quello di fornire le specifiche per un terminale "standard" di rete, cioè di un "modello di base" di un terminale, indipendente dalle macchine connesse.
Il NVT è un ambiente virtuale creato ai due estremi della connessione, dove operano rispettivamente il client e il server Telnet: naturalmente i due host possono essere di tipo diverso, con sistema operativo diverso e software Telnet diverso.



NVT definisce un terminale di rete virtuale che fornisce un'interfaccia standard ai sistemi remoti e quindi non è necessario utilizzare software specifici sia dal lato client che da quello server.

Per effettuare la connessione sono disponibili poche funzioni che vengono negoziate tra i due host in modo da avere un ambiente identico con caratteristiche comuni: abbiamo quindi due NVT identiche sulle due macchine, composte da
- caratteri ASCII 7 bits ai quali si aggiungono quelli del codice ASCII esteso;
- tre caratteri di controllo;
- uno schermo che scrive i dati in arrivo;
- una tastiera che produce i dati che spedisce attraverso la connessione Telnet e anche al proprio schermo (effetto "Echo").

Telnet ha definito in pratica uno standard affinché terminali di diverso tipo, con differenti caratteristiche e set di caratteri, potessero essere messi in comunicazione fra loro, utilizzando un'astrazione del terminale stesso, consentendo a host diversi di comunicare tra loro.

Ogni conversione di dati è un problema locale che non coinvolge Telnet.

Il principio di opzioni negoziate

Nella connessione tra due host diversi è possibile che essi possano fornire un insieme di funzionalità diverse, come servizi aggiuntivi rispetto a quelli di base comuni a tutti i NVT standard: per esempio alcuni software offrono una interfaccia più gradevole all'utente, sia nella forma, per esempio grafica, sia nella dimensione dello schermo per migliorarne la visualizzazione.
Queste funzionalità si traducono in termini di opzioni.

Il protocollo Telnet contempla questa possibilità e definisce la modalità per cui all'atto del collegamento i due soggetti cercano di comune accordo di trovare la migliore forma di "collaborazione".

Operativamente questa fase di dialogo, che prende il nome di negoziazione, avviene all'atto della connessione nel seguente modo:
- un host indica all'altro che intende usare una certa opzione;
- questo gli risponde o accettando quella opzione, o proponendo a sua volta una nuova opzione;
- se non si trova un accordo, si utilizza la modalità standard.

Se un host riceve una richiesta di disabilitazione di una opzione è naturalmente obbligato ad accettarla in quanto il suo interlocutore gli sta indicando l'impossibilità di comunicare in quella modalità.

Sostanzialmente abbiamo quattro tipi di richieste:
- vuole usare (DO)
- rifiuta di usare (DON'T)
- vuole che l'altra estremità usi (WILL)
- rifiuta che l'altra estremità usi (WON'T)

Ogni richiesta è seguita da un numero che indica il tipo di opzione, come per esempio:

```
WILL XXX
```

indica che un partner desidera usare l'opzione XXX;

```
DO XXX
```

e

```
DON'T XXX
```

saranno gli acknowledgments positivo e negativo.

Nel caso opposto, il comando

```
WON'T XXX
```

indica che un partner non desidera più usare l'opzione XXX.

Il comando

```
DO XXX
```

indica che un partner richiede all'altro partner di usare l'opzione XXX;

```
WILL XXX
```

e

```
WON'T XXX
```

saranno gli acknowledgments positivo e negativo.

Nel caso opposto, il comando

```
DON'T XXX
```

indica che un partner richiede all'altro partner di togliere l'opzione XXX.

Combinando proposta e risposta otteniamo le possibili situazioni riportate nella seguente tabella:

Richiesta	Risposta	Interpretazione
DO	WILL	L'emittente comincia usando l'opzione
	WON'T	L'emittente non deve usare l'opzione
WILL	DO	L'emittente comincia usando l'opzione, dopo aver inviato un *DO*
	DON'T	L'emittente non deve usare l'opzione
DON'T	WON'T	L'emittente segnala che ha disattivato l'opzione
WON'T	DON'T	L'emittente segnala che deve disattivare l'opzione

Esistono 255 codici di opzioni e la RFC 855 spiega come documentare ogni nuova opzione.

Per permettere agli hosts che non supportano alcune opzioni di creare una connessione Telnet, all'inizio del collegamento tutte le opzioni sono disabilitate; tutte le richieste di opzioni non supportate verranno rifiutate.

La simmetria del collegamento

Alla base del protocollo Telnet è stato posto il concetto di simmetria, cioè si è scelto un protocollo simmetrico in quanto molto semplice da realizzare: questo però può portare a inconvenienti dovuti proprio al fatto che non essendoci un host prevalente la fase di negoziazione delle opzioni potrebbe portare a un "ciclo infinito" di richieste nel caso di una errata interpretazione di una richiesta.

Per prevenire queste possibili situazioni indesiderate, l'NVT deve seguire queste regole:
- ogni interlocutore può richiedere un solo cambio nelle opzioni;
- non è possibile richiedere lo stato di una opzione;
- se viene inoltrata una richiesta per un'opzione già attiva, questa viene ignorata;
- deve essere inviato un ACK in risposta per ogni richiesta di cambio di stato;
- se viene richiesto il cambio di stato di una opzione che modifica il trattamento dei dati, questo deve essere inserito nel flusso di dati nel punto in cui gli stessi devono essere interpretati diversamente.

ESEMPIO 3

Supponiamo che l'host A debba spedire i caratteri X, Y, Z, CR, LF, 1, 2, ma che voglia spedire gli ultimi due, cioè 1 e 2, subito, senza aspettare il prossimo *end of line*; è possibile farlo in questo modo:

- prima si spediscono i caratteri X, Y, Z, CR, LF e si memorizzano in un buffer i caratteri 1 e 2;
- quindi si inviano due richieste di opzioni DO ECHO e DO SUPPRESS-GO-AHEAD per informare l'host B che l'output seguente sarà del tipo char-at-time;
- l'host B risponde con WILL ECHO e WILL SUPPRESS-GO-AHEAD;
- alla ricezione dei due ack, come conferma delle due richieste di opzione, si spedisce il carattere 1 e 2;
- ora ogni carattere, non appena verrà processato, verrà spedito.

Comandi e funzioni standard

Ogni comando di controllo inviato da un client a un server Telnet è composto da due parti: il codice di Escape detto anche IAC (Interpret As Command) e il codice del comando.

I codici di Escape hanno lo scopo di distinguere i comandi dai dati dato che il valore di escape viene codificato con il valore decimale 255 per indicare che a esso segue un ottetto di codice di controllo. La seguente tabelle riporta alcuni classici comandi di Telnet:

Nome	Codice	Significato
SE	240	Fine dei parametri della subnegoziazione
NOP	241	Nessuna operazione
Data Mark	242	Una porzione del Synch - deve essere accompagnato da TCP Urgent notification
Break	243	Carattere BRK dell'NVT
Interrupt Process	244	La funzione IP
Abort output	245	La funzione AO
Are You There	246	La funzione AYT
Erase character	247	La funzione EC
Erase Line	248	La funzione EL
Go ahead	249	Il segnale GA
SB	250	Indica che ciò che segue è la subnegoziazione dell'opzione indicata
WILL (codice opzione)	251	Indica il desiderio o la conferma di usare l'opzione indicata
WON'T (codice opzione)	252	Indica il rifiuto di usare o di continuare a usare l'opzione indicata
DO (codice opzione)	253	Indica la richiesta che il partner usi l'opzione indicata
DON'T (codice opzione)	254	Indica la richiesta che il partner termini di usare l'opzione indicata

ESEMPIO 4

Per richiedere l'interruzione di un programma si invia la sequenza:
255 (IAC) seguito da 244 (IP).

Per richiedere un Break la sequenza di caratteri è la seguente:
255 (IAC) seguito da 243 (BRK).

I comandi riguardanti la negoziazione delle opzioni sono una sequenza di tre byte; il terzo byte sarà il codice dell'opzione referenziata.

ESEMPIO 5

Per spedire il comando DO TRANSMIT-BINARY la sequenza di caratteri è la seguente:
255 (IAC), 253 (DO) e 0 (opzione TRANSMIT-BINARY).

La (non) sicurezza di Telnet

Quando Telnet è stato sviluppato all'inizio degli anni '80 la maggior parte degli utenti delle reti apparteneva a dipartimenti di università, a centri di ricerca privati o governativi oppure ad aziende private, ambienti dove non era importante la sicurezza dato che le informazioni venivano trasmesse sempre all'interno delle singole organizzazioni.
Con l'avvento delle reti remote e la crescita esponenziale del numero di persone connesse in internet, i limiti di Telnet lo rendono praticamente inutilizzabile a causa dei due seguenti problemi:

- Telnet non cripta i dati inviati ed è quindi banale catturare i dati scambiati e usare la password da parte di malintenzionati dato che neanche queste vengono criptate!
- manca uno schema di autenticazione che renda sicura la comunicazione tra due host e non intercettabile.

> In Internet, dove è importante utilizzare sistemi sicuri, Telnet non dovrebbe usato.

Infatti è sufficiente un qualunque software di packet sniffing (come tcpdump e wireshark) per intercettare i pacchetti scambiati da due host che stanno comunicando tramite Telnet, leggere in chiaro tutte le informazioni e ottenere qualsiasi cosa venga scambiata, quindi impossessarsi facilmente dei nomi utente e rispettive password.

Nel 1998 Telnet fu praticamente abbandonato a favore di un nuovo protocollo più sicuro, l'SSH, che, poiché offre tutte le funzioni di Telnet più una sicura criptazione e un'autenticazione a chiave pubblica (viene descritto nella sezione riservata alla sicurezza nel volume 3 di questo corso) è diventato uno standard di fatto per l'amministrazione remota.

> I principali browser non integrano nella loro interfaccia Telnet appunto per motivi di sicurezza e quindi non è possibile visualizzare dall'interno di essi schermate in emulazione terminale: per utilizzare Telnet bisogna richiamare all'interno di una pagina WWW un programma aggiuntivo (come per esempio PuTTY, che è un client Telnet con crittografia che utilizza SSH).

Utilizzo di Telnet

Ma come viene utilizzato oggi Telnet, se è così vulnerabile?
L'uso principale di Telnet avviene nelle fasi di debug di servizi di networking come i server SMTP e HTTP, in quanto rappresenta un modo semplice per mandare comandi al server ed esaminare le risposte, e nei giochi ◀ **MUD** ▶ in rete.



◀ **MUD** A MUD (Multiple User Dimension, Multiple User Dungeon, or Multiple User Dialogue) is a computer program which users can log into and explore. Each user takes control of a computerized persona/avatar/incarnation/character.
You can walk around, chat with other characters, explore dangerous monster-infested areas, solve puzzles, and even create your very own rooms, descriptions and items. ▶

Anche nella posta elettronica Telnet trova varie utilizzazioni, come quella di offrire l'agevole lettura della corrispondenza sulla propria mailbox, cancellarla oppure spedire missive elettroniche dato che, come vedremo in seguito, normalmente l'accesso alla propria casella di posta elettronica viene fatto in modo non sicuro.
Viene anche utilizzato per accedere alla banche dati libere presenti in tutto il mondo, come per esempio quelle riportate nella seguente tabella:

Nome host	Ente/tipo di archivio	Login
access.usask.ca	Hytelnet database di cataloghi di biblioteche.	hytelnet
envirolink.org	Database e sistema di conferenze sull'ambiente, Pittsburg.	gopher
callcat.med.miami.edu	Università di Miami, gestisce elenco di strutture specializzate per malati di AIDS, si possono contattare dottori e strutture specializzate.	library
hpcvbbs.cv.hp.com	Hewlett-Packard, per chiedere informazioni.	
martini.eecs.umich.edu 3000	Geographic Name Server Università del Michigan.	
ipac.caltech.edu	Dati su 100.000 galassie.	ned
forsy.thetn.stanford.edu	Università di Stanford, documenti riguardanti Martin Luther King.	socrates
madlab.sprl.umich.edu 3000	Per scaricare immagini satellitari e radar metereologici.	

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 L'acronimo NVT deriva da:

a) Network Virtual Transport
b) Network Virtual Terminal
c) Natural Virtual Transport
d) Natural Virtual Terminal

2 Quali delle seguenti affermazioni relativa al NVT è falsa?

a) Si può accedere a tutti gli host usando la stessa interfaccia (minimale).
b) Consente di collegarsi a un host remoto e operare come se si fosse in locale.
c) L'input è gestito dalla tastiera locale.
d) L'output è visualizzato sul monitor locale.
e) L'echo è visualizzato sul monitor remoto.
f) La computazione è effettuata sull'host remoto.

3 Il protocollo Telnet si basa su tre concetti fondamentali (indica quello errato):

a) il paradigma del terminale virtuale di rete
b) il principio di opzioni negoziate
c) il principio di bidirezionalità
d) la simmetria del collegamento

4 Nella negoziazione abbiamo quattro tipi di richieste (indica quella errata):

a) DO
b) WIN
c) DON'T
d) WON'T

5 La sequenza DON'T WON'T:

a) è errata dato che una richiesta non è contemplata
b) indica che l'emittente ha disattivato l'opzione
c) indica che l'emittente non deve utilizzare l'opzione
d) è impossibile
e) indica al ricevente di disattivare l'opzione
f) indica che l'emittente deve disattivare l'opzione

6 Per prevenire situazioni di loop l'NVT deve seguire queste regole (indica quella errata):

a) ogni interlocutore può richiedere un solo cambio nelle opzioni
b) non è possibile richiedere lo stato di una opzione
c) se viene inoltrata una richiesta per una opzione già attiva, questa viene disattivata
d) deve essere inviato un *ack* in risposta per ogni richiesta di cambio di stato

Test vero/falso

1 Le versioni odierne dei sistemi operativi hanno istallato di default il client Telnet. V F
2 Le versioni odierne dei sistemi operativi hanno attivato di default il client Telnet. V F
3 Per avviare una sessione Telnet occorre indicare la porta alla quale connettersi. V F
4 Il NVT è un ambiente virtuale creato alla estremità client della connessione. V F
5 NVT definisce un terminale di rete virtuale che fornisce un'interfaccia standard. V F
6 Le conversioni di dati sono problemi locali che non coinvolgono Telnet. V F
7 La fase di negoziare le opzioni potrebbe portare a un "ciclo infinito" di richieste. V F
8 Al codice di escape è stato assegnato il valore decimale 155. V F
9 I comandi di negoziazione delle opzioni sono formati da una sequenza di due byte. V F
10 SSH offre una sicura criptazione e un'autenticazione a chiave pubblica. V F

Domande aperte

1 A cosa serve il protocollo Telnet?

2 Che cos'è un NVT?

3 A cosa serve l'NVT?

4 Cosa sono lo schermo e la tastiera dell'NVT?

5 A cosa serve la negoziazione delle opzioni?

6 Come si negoziano le opzioni?

7 In che senso la negoziazione delle opzioni sfrutta la simmetria?

8 La visione simmetrica delle opzioni può portare a malfunzionamenti o a loop?

9 Cosa sono le funzioni standard del protocollo Telnet?

10 Le funzioni standard devono essere implementate?

LEZIONE 3

WEB E HTTP

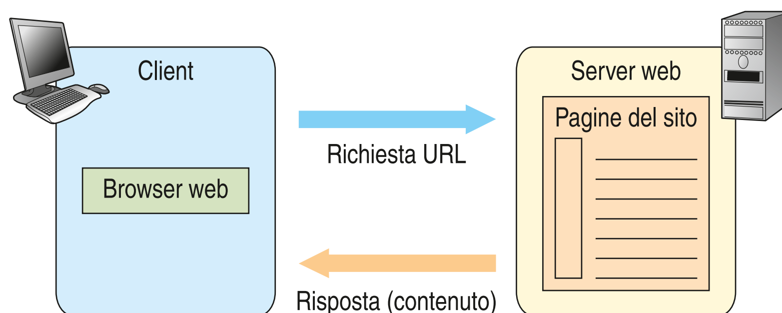
IN QUESTA LEZIONE IMPAREREMO...

- l'architettura del Web
- il protocollo HTTP
- il formato del messaggio HTTP

Il World Wide Web

WWW (World Wide Web) significa "ragnatela intorno al mondo" ed è un insieme di pagine multimediali, documenti testuali, audio e video, collegati tra loro, entro i quali ci si può spostare con diverse modalità. In sostanza, è l'insieme delle pagine ipermediali di Internet, cioè è un ipertesto multimediale distribuito.

Lo schema che definisce il funzionamento del WWW è presentato nella figura che segue.





IPERTESTO

Un ipertesto è un testo la cui struttura è reticolare, invece che semplicemente lineare o gerarchica: si compone di nodi ("pagine" o page) e collegamenti (link).
I link sono collegati a particolari parole, hotspot o hotword, che, se selezionate, indirizzano verso un'altra pagina in modo da permettere di raggiungere un altro punto dell'ipertesto o una nuova risorsa.

Una pagina, detta home-page, viene inviata come risposta, all'atto del collegamento, a un indirizzo specifico e per essere visualizzata sull'host che ne ha fatto la richiesta è necessario l'uso di un ◀ browser ▶, un programma client per la visualizzazione dei documenti e per la navigazione in rete.



◀ **Browser** Significa letteralmente "sfogliatore" e non è altro che un programma in grado di "locare" una pagina in Internet e di interpretarne attraverso un parser interno le righe di codice sorgente scritte in HTML, fornendo all'utente una linea di comando dove poter digitare gli indirizzi IP. ▶

Il codice sorgente delle pagine Web che viene interpretato è scritto in un linguaggio di "contrassegno" quale HTML, che nelle ultime versioni è anche in grado di mandare in esecuzione pseudoeseguibili scritti in Java (Applet) o di interpretare programmi incapsulati internamente – detti script –, quali per esempio VBscript, Jscript o il più conosciuto JavaScript. I browser più diffusi sono certamente Internet Explorer (a volte denominato IE per brevità) di Microsoft e FireFox oltre ad altri meno noti (Opera, Chrome e Safari).

Il client fornisce a un server le richieste per un determinato oggetto. Il server risponde con i dati richiesti inviandoli in formati standard integrandoli col codice HTML. Questo è un modello di carattere generale, tuttavia sarà importante ricordare questo schema di comunicazione quando verranno discussi i linguaggi di scripting lato server (PHP e ASP).

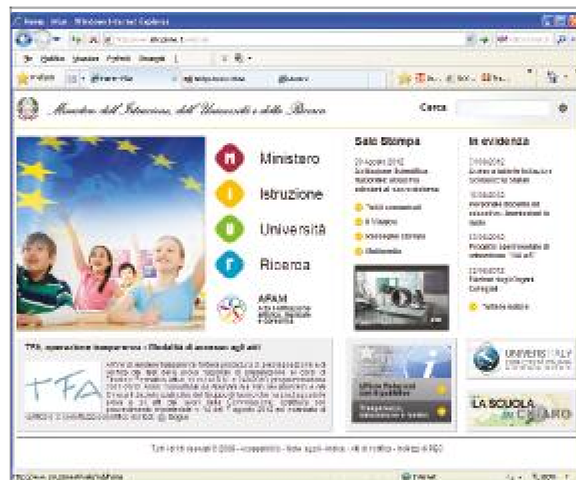
L'architettura del Web

All'interno della rete Internet ciascuna risorsa è identificata in modo univoco mediante un indirizzo URI (Universal Resource Identifier). L'URI non è altro che un indirizzo univoco di una particolare risorsa nella rete, rappresenta la risorsa che si vuole raggiungere. Per capire il funzionamento del Web vediamo un piccolo esempio che aiuta a comprendere cosa accade quando un utente si vuole collegare a un certo indirizzo URI, in questo caso al sito http://www.istruzione.it.

Si scrive nell'apposita barra del browser l'indirizzo http://www.istruzione.it.



- il browser recupera le informazioni secondo i meccanismi previsti per le risorse identificate con lo schema URI-HTTP;
- il server che contiene le pagine Web del sito www.istruzione.it risponde alla richiesta inviando le informazioni che si vogliono ottenere;
- il browser interpreta la risposta, identificando il tipo di documento che viene trasmesso (content-type) e a sua volta invia altre richieste per recuperare le informazioni collegate, quali per esempio le immagini relative alle notizie dell'ultim'ora;
- il browser mostra le informazioni ricevute, che includono a loro volta collegamenti ipertestuali.



L'URI comprende tutto quanto è necessario per l'indirizzamento e la localizzazione dei file. Esso infatti include la localizzazione del protocollo, l'indirizzo Internet e il nome del file, ed eventualmente la localizzazione interna a quel particolare documento. Questo argomento verrà approfondito più avanti.

URL (Uniform Resource Locator)

Generalmente si sente parlare di URL (Uniform resourse locator) e non di URI, ma la differenza tra i due riferimenti a indirizzi Internet sta nel fatto che l'URL è un particolare tipo di URI e un URI può identificare oggetti anche non correlati ai protocolli Internet esistenti (si pensi per esempio al caso di about:netscape).

Ogni documento Internet è individuato da un indirizzo URL che lo identificano in modo univoco, che è formato da:

- protocollo per la connessione (HTTP per pagine web);
- nome simbolico (o indirizzo IP) del server;
- pathname del file sul server.

> In pratica un URI è un URL più generale che non fa riferimento a un preciso schema di denominazione.

ESEMPIO 6

http://www.lorusso.it/Sistemi/Volume2/index.html

È anche possibile inoltre indicare la porta a cui connettersi e passare dei parametri alla risorsa richiesta:

http://www.lorusso.it:8083/Sistemi/Volume2/page.php?param=value

Vedremo che l'associazione tra il nome dell'host e il suo indirizzo IP viene effettuata da una apposita applicazione, DNS (Domain Name System).

L'architettura di funzionamento del Web può essere espressa mediante tre termini assai importanti che ne descrivono le caratteristiche: identificazione, interazione e formato.

L'identificazione

Mediante un indirizzo (URI o URL) le risorse presenti nella rete possono essere univocamente identificate.

L'interazione

I browser comunicano mediante protocolli standard che permettono l'interazione tramite l'interscambio di informazioni strutturate secondo una sintassi e una semantica precise. Infatti quando si clicca su un link o ci si collega a un indirizzo Web, si dice al browser di ricercare una risorsa identificata da quell'indirizzo.

Il formato

Nel protocollo di comunicazione sono previste informazioni aggiuntive, chiamate meta-informazioni che permettono di identificare il tipo di documento da trasferire o da interpretare.

> Nel caso di pagine HTML il formato previsto ò di tipo Content-type:txt/html: questo è il formato standard dei documenti scritti in linguaggio di contrassegno e tutti i browser sono in grado di interpretare pagine secondo questo formato.
> Maggiori e interessanti informazioni sul funzionamento degli URI e delle architetture Web si possono trovare all'indirizzo Internet della società W3C (abbreviazione di www Consortium): http://www.w3.org/TR/Web arch/.



Zoom su...

LA SINTASSI URL

La sintassi prevede che venga indicato il protocollo per primo, quindi le informazioni del dominio, poi l'eventuale country-code, quindi le eventuali directory e infine il file che contiene i dati.

protocollo usato: sottodominio"n".sottodominio"n-1"...dominio/directory/file

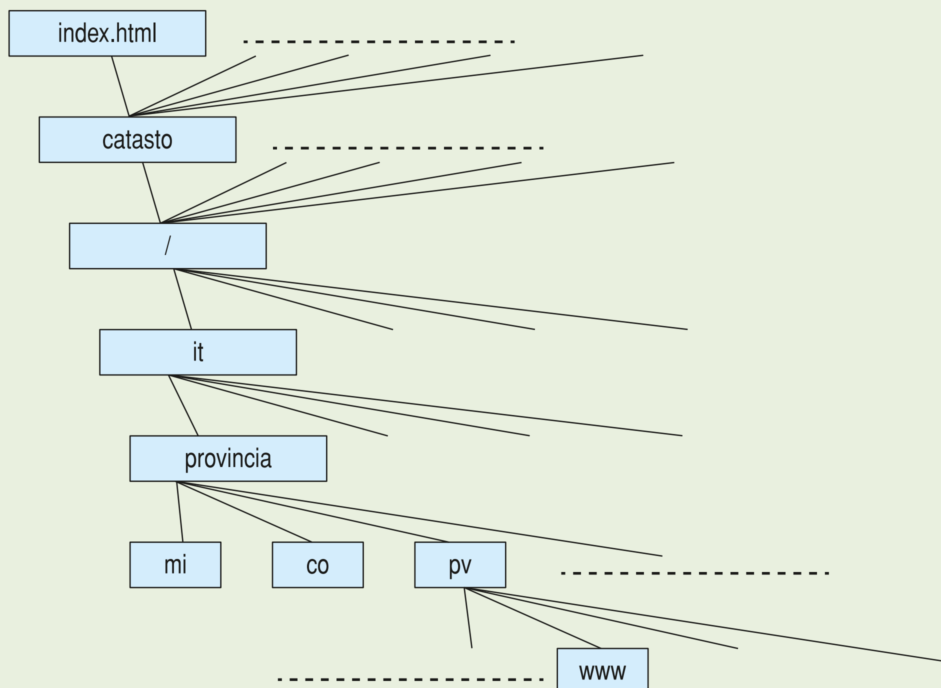
Il dominio può possedere anche dei sottodomini, per esempio nell'indirizzo

http://www.mi.provincia.it

mi è un sottodominio di provincia, oppure nell'url

http://www.mi.provincia.it/cataloghi/anni/2012/miocatalogo.htm,

cataloghi è la directory del server di dominio che contiene la directory anni che a sua volta contiene la directory 2012 che a sua volta contiene il file. www invece indica il web server, infatti può essere sostituito con il nome della macchina che lo ospita, per esempio l'url "http://www.lombardia.regione.it" rispetto a "http://italia.lombardia.regione.it", presenta "italia" che non è altro che la macchina che fisicamente ospita il web server.
I domini vengono appunto suddivisi gerarchicamente: più nomi di macchine fanno riferimento a un dominio, più nomi di domini fanno riferimento a un dominio di alto livello. Per esempio l'url "http://www.pv.provincia.it/catasto/index.html" può essere schematizzato come segue:



Ma che cos'è un dominio di alto livello? Si tratta di un dominio che identifica la tipologia e la topologia di un sito.

Vediamo una tabella che ne descrive alcuni:

.com	per organizzazioni a carattere commerciale	**.gov**	per enti governativi americani
.edu	per organizzazioni di ricerca americane	**.uni**	solo per università americane
.net	per organizzazioni che forniscono servizi di rete	**.org**	per società o organizzazioni non commerciali
.it	Italia	**.fr**	Francia
.de	Germania	**.uk**	Gran Bretagna
.mn	Mongolia	**.my**	Malesia
.hk	Hong Kong	**.tr**	Turchia
.cu	Cuba	**.ir**	Iran
.cg	Congo	**.ch**	Svizzera
.biz	Business	**.tv**	Televisioni

Da come si può vedere in questa tabella, che riassume solo alcuni suffissi di dominio, sono divisi in country-code e in society-code, i primi nazionali e i secondi invece legati a enti, aziende o associazioni. Tuttavia uno dei problemi che ultimamente è venuto alla ribalta è proprio legato a come questi suffissi non siano più sufficienti e vi sia la necessità di nuove adozioni.
Anche per gli indirizzi di posta elettronica deve essere indicato un URL, che è composto dai seguenti componenti:

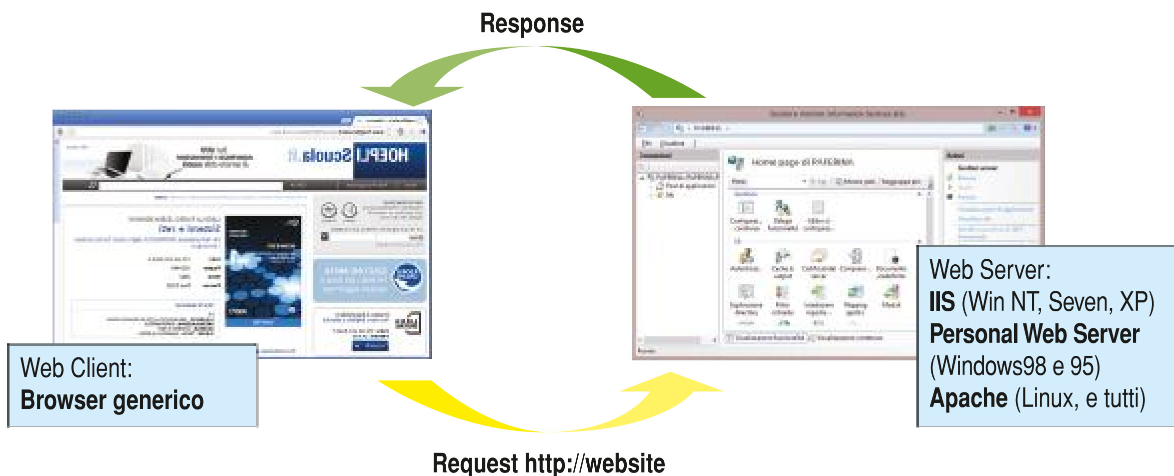
mailto:username@dominio

Per esempio nell'url "mailto:zio.pino@co.provincia.it" il dominio è composto da un host principale (provincia.it) e da un sottohost denominato co.

Il protocollo Hyper-Text Transfer Protocol (HTTP)

Il World Wide Web per il trasferimento di dati ipertestuali si basa sul protocollo applicativo Hyper TextTransfer Protocol (HTTP) che, come tutti i servizi di rete utilizza l'architettura *client/server*.

Il protocollo http, è definito in RFC 1945 (versione 1.0) ed RFC 2616 (versione 1.1), è basato sullo scambio di messaggi tra client e server. Come si vede nella figura, abbiamo a sinistra il browser (client) e a destra il server (rappresentato in questo caso da un *IIS – Internet Information server* – che non è altro che un server Web utilizzato in tutti i sistemi operativi della Microsoft):



La comunicazione tra client e server avviene secondo il protocollo TCP/IP usando gli indirizzi IP dei computer che ospitano client e server.

I messaggi con il protocollo http possono essere separati nelle due categorie request e response, cioè richiesta e risposta, e sono un insieme di righe di caratteri ASCII terminate da CR+LF. I due messaggi hanno lo stesso formato, secondo lo standard RFC822, e sono entrambi definiti dal protocollo.

Il meccanismo che sta alla base della comunicazione attraverso il protocollo HTTP è il seguente:
- si apre una connessione TCP tra client e server;
- il browser richiede una risorsa al server http (web server);
- il server risponde (se possibile, fornendo la risorsa richiesta);
- si chiude la connessione (da HTTP1.1 non si chiude automaticamente ogni volta la sessione, ma rimane aperta per dare maggiori possibilità di trasmissione dati).

Vediamo adesso come si apre la connessione tra client e web server in una tipica "conversazione" tramite protocollo http, per esempio per la seguente richiesta:

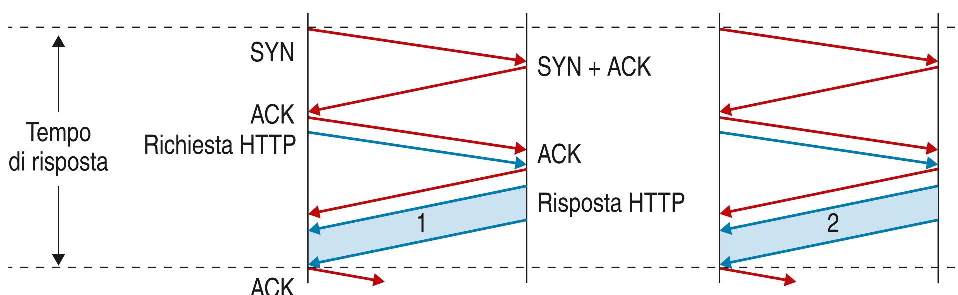
http://alinet.miur.unibc.it/sistemi/itisVol2/RetiLaB/index.html

1. Il browser del client analizza l'URL e ne estrae il dominio;
2. il client HTTP inizia una connessione TCP verso il web server alinet.miur.unibc.it sulla cui porta 80 il server HTTP è in ascolto;
3. il client HTTP invia una richiesta (GET) al server attraverso il socket associata alla connessione TCP stabilita al punto 1; nella richiesta specifica il file /sistemi/itisVol2/RetiLaB/index.html;
4. il server riceve la richiesta, incapsula l'oggetto specificato nella risposta HTTP e invia il messaggio al client attraverso il socket;
5. il server HTTP invia quindi al TCP la richiesta di chiusura della connessione, che avverrà solo dopo che il messaggio di risposta è stato riscontrato dal client;
6. la connessione TCP si conclude; il messaggio indica che l'oggetto è un file HTML; il client lo estrae e trova i riferimenti agli oggetti referenziati (immagini, fogli di stile, oggetti Javascript…);
7. i passi precedenti vengono ripetuti per ciascuno degli oggetti referenziati (eventualmente aprendo più connessioni in parallelo).

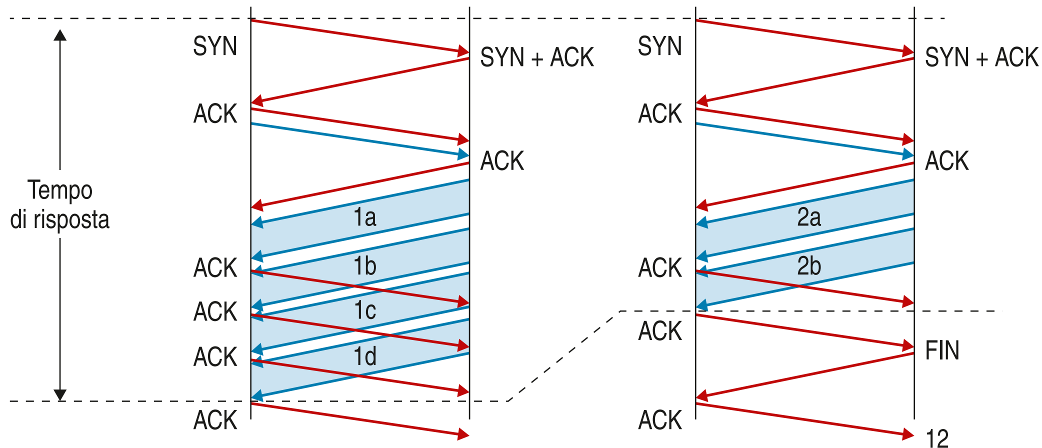
> La connessione creata da HTTP 1.0 è di tipo non permanente: si limita alla trasmissione di una singola pagina HTML. La modalità di default per HTTP 1.1 è invece di tipo permanente: il server non chiede la chiusura della connessione TCP se non dopo che è scaduto un time-out durante il quale la connessione non è utilizzata.

Analizziamo i tempi necessari per ricevere il file base HTML nel caso di connessione non permanente:

A. 1 RTT (Round Trip Time) per l'instaurazione della connessione TCP;
B. 1 RTT per l'invio della richiesta e l'arrivo della risposta;
C. tempo di trasmissione del file HTML al client.

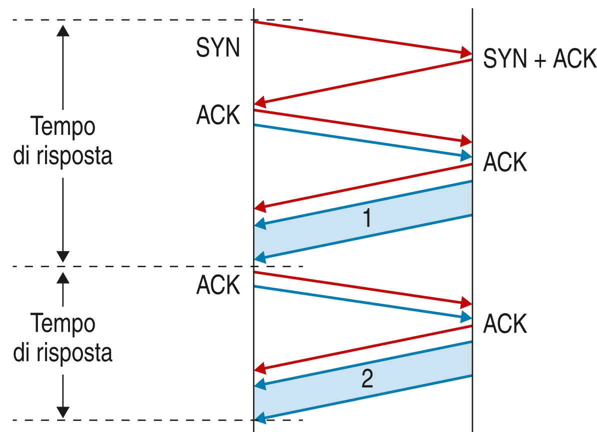


Per messaggi lunghi abbiamo due possibili situazioni come quelle riportate nella figura seguente:



Per ricevere un oggetto (tranne il primo) nel caso di connessione permanente occorre:

- **A** 1 RTT;
- **B** tempo di trasmissione del file HTML.



Si può osservare come il secondo messaggio non abbia bisogno della fase di istaurazione della connessione e quindi è evidente il risparmio di tempo dell'HTTP 1.1.

Il formato del messaggio HTTP

Il formato dei messaggi HTTP è formato da una start-line (riga di richiesta/risposta), una HEADER (intestazione HTTP) e da un BODY (corpo http, che può anche essere omesso). Vediamoli più in dettaglio.

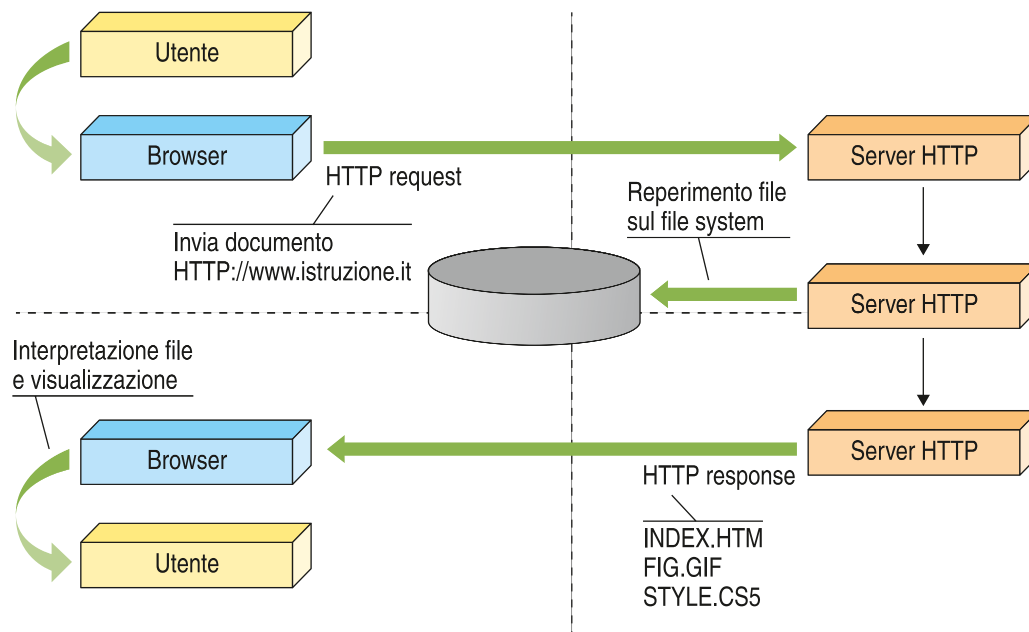
La prima riga della richiesta contiene 3 elementi, un comando chiamato metodo (GET e POST sono i più usati, poi ne vedremo le differenze, ma esistono anche HEAD, PUT, DELETE, TRACE, CONNECT, OPTIONS, per maggiori chiarimenti guardare il sito www.rfc.net), il secondo è il percorso del server della richiesta del client e poi la versione del protocollo http usata (di solito 1.1).
Nella intestazione (HEADER) compaiono invece i dati che riguardano il tipo di documento che il client riceve dal server, il tipo di browser che ha richiesto la pagina, la data e alcune informazioni generali sulla configurazione del sistema richiedente. Il corpo (BODY) del messaggio di richiesta da parte del client, può essere omesso, verrà usato invece dal server per rispondere con la pagina

HTML vera e propria. Conterrà i dati di un eventuale form inviato al server SOLO se è stato usato il metodo POST. Nella riga risposta del server invece compaiono solo 2 informazioni: il numero delle versioni dell'HTTP e un codice di risposta tra i seguenti:

Classe	Descrizione
100-199	Risposta in fase di elaborazione
200-299	Successo
300-399	La richiesta non è stata evasa in quanto le informazioni richieste sono state spostate
400-499	La richiesta è incompleta, errata o impossibile (è tipico l'errore **404**: server non trovato)
500-599	Errore nel server, con richiesta però valida

Nell'intestazione della risposta compaiono 3 dati: il primo di tipo generale (per esempio data e ora di risposta), il secondo è l'entità (per esempio data e ora di ultima modifica dei dati richiesti) e il terzo è la richiesta (informazioni sul server e su come gestisce la risposta, se i dati sono text o html ecc.). Il corpo della risposta contiene la pagina HTML vera e propria.

Vediamo lo schema logico di quanto appena detto:



ESEMPIO 7

Vediamo adesso tre esempi di richiesta/risposta.

1 Esempio di richiesta.

```
GET /sistemi/itisVol2/RetiLaB/index.html HTTP/1.1     */* metodo, file, versione
Accept:                                               */* contenuto accettato
Accept-Language: en-us                                */* preferenza linguistica
If-Modified-Since: Wed, 16 Jan 2012 12:37:40 GMT      */* ultima versione nella cache
User-agent:Mozilla/4.0                                */* tipo di browser
Host: alinet.miur.unibc.it                            */* host
Connection: Keep-Alive                                */* connessione permanente
```

2 Esempio di risposta positiva.

```
HTTP/1.1 200 OK
Date: Wed, 22 Mar 2013 17:37:44 GMT
Content-Length: 12692
Content-Type: text/html
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Mon, 01 Mar 2013 18:02:44 GMT
<html>
…
qui c’è il testo HTML della pagina richiesta (12692 byte)
…
</html>
```

3 Esempio di risposta negativa (il file richiesto non esiste).

```
HTTP/1.1 404 Not Found
Date: Wed, 02 Mar 2013 17:38:37 GMT
Content-Length: 1067
Content-Type: text/html
Server: Apache/2.0.40 (Red Hat Linux)
<html>
…
Object not found
…
Error 404
…
</html>
```

Definizioni dei metodi

HTTP mette a disposizione del client un insieme di metodi per inviare le richieste al server: questi metodi sono a tutti gli effetti dei comandi che il client invia al server. Vediamoli in sintesi.

OPTION

Il metodo OPTION rappresenta una richiesta di informazioni inerenti alle opzioni di comunicazione disponibili sul canale definito dalla Request-URI.

GET

Il metodo GET richiede una risorsa (informazioni, file, documenti, pagine HTML) al server localizzato con una Request-URI.

HEAD

Il metodo HEAD è identico al GET eccetto per il fatto che il server non deve restituire il corpo del messaggio, ma solamente l’header. Il metodo viene usato spesso per testare la accessibilità e le recenti modifiche di links ipertestuali.

POST

Con il metodo POST il client può spedire al server informazioni organizzate con una serie di coppie *nome=valore* che corrispondono all’input del programma indicato nella request-URI.

PUT
Il metodo PUT serve per inviare file al server (upload di file da parte del client): se il file esiste in corrispondenza all'URI specificato, questo verrà considerato come un aggiornamento della versione preesistente.

DELETE
Il metodo DELETE richiede che il server ricevente cancelli il file all'indirizzo specificato dal Request-URI. Il client non ha nessuna garanzia che l'operazione abbia esito positivo in quanto è necessario che esso possieda i diritti per eseguire questa operazione.

TRACE
Il metodo TRACE richiama un loopback remoto a livello dell'applicazione del messaggio richiesto: si tratta di un "ping" che verifica quali dati il server Web riceve dal client.

CONNECT
Questo metodo viene usato per instaurare una semplice connessione con un proxy server.

■ Proxy server

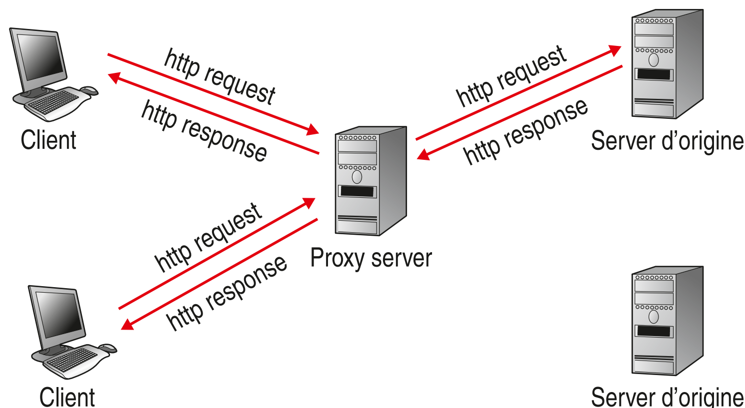
I ◀ **proxy server** ▶ sono dei software che in un'architettura client-server si frappongono tra il server e il client con funzione di ponti nella rete: al proxy viene delegato il compito di scaricare le informazioni e di memorizzarle su una memoria di massa in modo da poter essere restituite il più rapidamente possibile durante successive richieste identiche.



◀ **Proxy server** Da Wikipedia: "In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource available from a different server. The proxy server evaluates the request as a way to simplify and control their complexity." ▶

Il proxy ha in genere una connessione in banda larga con la rete, quindi molto più veloce dei singoli host, e svolge la funzione di cache (Web Caching), cioè memorizza le pagine usate più recentemente e frequentemente dagli utenti e quando l'utente le richiede, vengono inviate dal proxy e non dal sito da cui provengono: quindi l'obiettivo dell'utilizzo dei proxy server è quello di migliorare la velocità di risposta della rete cercando di soddisfare le richieste del client senza coinvolgere il server originale.

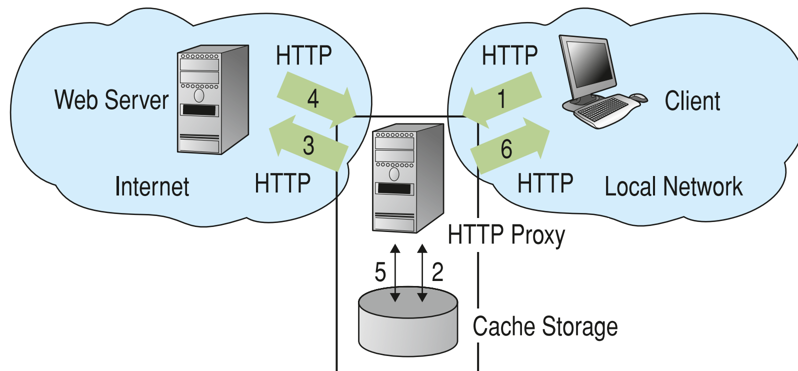
Il client manda tutti gli HTTP request al web cache:
- se l'oggetto richiesto è nel web cache, il web cache spedisce l'oggetto;
- altrimenti il web cache richiede l'oggetto dal server d'origine e successivamente lo spedisce al client.



Vediamo una sequenza completa di funzionamento di un proxy come descritto nell'RFC 2616:

1. il client (browser) richiede una risorsa (URL);
2. il proxy intercetta la richiesta;
3. il proxy controlla la propria memoria cache per vedere se può rispondere al client direttamente oppure se deve richiedere la risorsa (URL) al server finale;
4. il server finale risponde al proxy;
5. il proxy aggiorna la propria memoria cache con la risposta del server;
6. il proxy crea un nuovo pacchetto e invia la risposta al client.

Nella figura seguente è mostrata la logica del funzionamento di un proxy HTTP.



> Con il Web Caching, se il web cache è "vicino" al client (es. nella stessa rete), i tempi di risposta sono inferiori; se i server sono distanti si ottiene una diminuzione del traffico (sappiamo che i canali esterni alla rete del ISP locale/istituzionale sono spesso colli di bottiglia).

Inoltre i proxy permettono l'accesso a una rete a più macchine e garantiscono l'anonimato nella navigazione sul web in quanto nascondono l'indirizzo IP del client che ha richiesto una pagina.
Quando un host si collega a Internet, il provider gli assegna un indirizzo IP (es. 82.210.62.18) al quale vengono associate le generalità del client che naviga con una sorta di "targa" grazie alla quale è rintracciabile per ogni evenienza e a ogni visita di un sito Web: il server che lo ospita leggerà l'indirizzo IP del computer e lo registrerà nel suo database.
Navigando tramite un proxy è possibile navigare anonimamente celando l'indirizzo IP: i siti che vengono visitati continueranno a ricevere informazioni sull'indirizzo, ma non quelle del client bensì quelle del server proxy (anche i cookies finiranno sul proxy e non sul computer).



Zoom su...

NAVIGAZIONE ANONIMA

Chi vuole navigare sul web servendosi di un proxy per rimanere anonimo e non possiede un proprio proxy server può utilizzare dei proxy server accessibili sul web: alcuni sono gratuiti, altri a pagamento, e si differenziano per servizi offerti e velocità nella navigazione.
Esistono molti siti che pubblicano liste di proxy server, eccone alcuni:

- NoTrace: dopo avere compilato il campo captcha è possibile accedere a una lista dei proxy accurata e ricca di dettagli che comprende più di 28000 server;
- Prospector: espone una lista di proxy HTTP gratuiti;
- SamAir: offre un'ampissima lista di proxy HTTP, con informazioni sui servizi offerti.

I server proxy possono anche essere classificati secondo il livello di anonimato che garantiscono:
- Trasparent Proxy Server: proxy trasparenti, anonimato minimo;
- Anonymous Proxy Server: proxy meno veloci, ma con migliore livello di anonimato;
- Elite Proxy Server: proxy con un elevato livello di anonimato.

Sono attualmente presenti oltre 40.000 Proxy Server.

I cookies

Il protocollo HTTP non permette che il server contattato "riconosca" un utente che si è già collegato, dato che ogni pagina inviata richiede una connessione TCP indipendente; inoltre dato che il protocollo HTTP è state-less e quindi le richieste dei client non lasciano alcuno stato nel server, è necessario memorizzare le informazioni della sessione sull'applicazione Web direttamente nel Web Browser del visitatore.

I ◀ **cookies** ▶ sono una parte fondamentale del protocollo HTTP che permettono di riconoscere un utente.

Un cookie è un file che viene memorizzato nel computer del client dal sito web che viene visitato, contenente informazioni sulle pagine visitate dall'utente: quando l'utente si ricollega, il contenuto di tale file viene mandato al server, che lo analizza e ne estrae informazioni che ha precedentemente memorizzato in modo da presentare le informazioni personalizzate in base alle esigenze del visitatore (per esempio la lingua preferita, i dati di login per non richiedere ripetutamente le credenziali di accesso ad aree riservate, la personalizzazione della visita con la visualizzazione delle preferenze già definite dall'utente in accessi precedenti ecc.).

In alcuni browser ogni cookie corrisponde a un piccolo file mentre per esempio in Firefox tutti i cookie sono memorizzati in un singolo file.

◀ **Cookies** HTTP cookies provide the server with a mechanism to store and retrieve state information on the client application's system: a cookie is a small file of letters and numbers placed by a website onto a user's computer when he or she accesses the website. ▶

Per poter scrivere i cookies sul computer dell'utente il server ha necessariamente bisogno della autorizzazione del client che deve abilitare tra le opzioni di configurazione del suo browser e di protezione del suo computer la possibilità di memorizzare i cookies da parte di altri host.

HTTPS: Secure HyperText Transfer Protocol (cenni)

Il normale traffico effettuato attraverso il browser col protocollo HTTP non pone in relazione i dati tra le sessioni precedenti e le successive e quindi rende di fatto "indipendente" ogni coppia di operazioni request-responce.
Supponiamo di voler effettuare un acquisto su un sito web e di dover quindi inviare informazioni riservate e personali: prima di arrivare a destinazione, i nostri dati, il numero della carta di credito e qualsiasi altra informazione ci riguardi, attraversano un numero imprecisato di nodi e sono visibili chiaramente con l'utilizzo di un semplice programma di ◀ **packet sniffer** ▶.

◀ **Packet sniffer** Packet sniffer are tools or utilities use to monitor and capture individual packets on the network to troubleshoot network problems or to find out what an attacker might see. ▶

La soluzione di questo problema è stata realizzata con la definizione del protocollo HTTPS descritto nella RFC 2965, che incapsula HTTP in una connessione cifrata con il Web Server di destinazione in modo da rendere più complesso, o per lo meno più macchinoso, un attacco da parte di malintenzionati.

Sintatticamente il protocollo HTTPS è identico al protocollo impiegato per la normale navigazione in rete e impiega oltre a i protocolli TCP e HTTP, un ulteriore livello chiamato SSL (Secure Sockets Layer) che prende i dati in entrata e li cripta attraverso un algoritmo matematico che li rende praticamente indecifrabili.
I dati transitano sulla porta 443 anziché 80.

Lo studio della sicurezza, della crittografia e della normativa relativa a tutti i metodi per garantire l'integrità dei dati e dei sistemi saranno oggetto di studio del quinto anno di corso.



Zoom su...

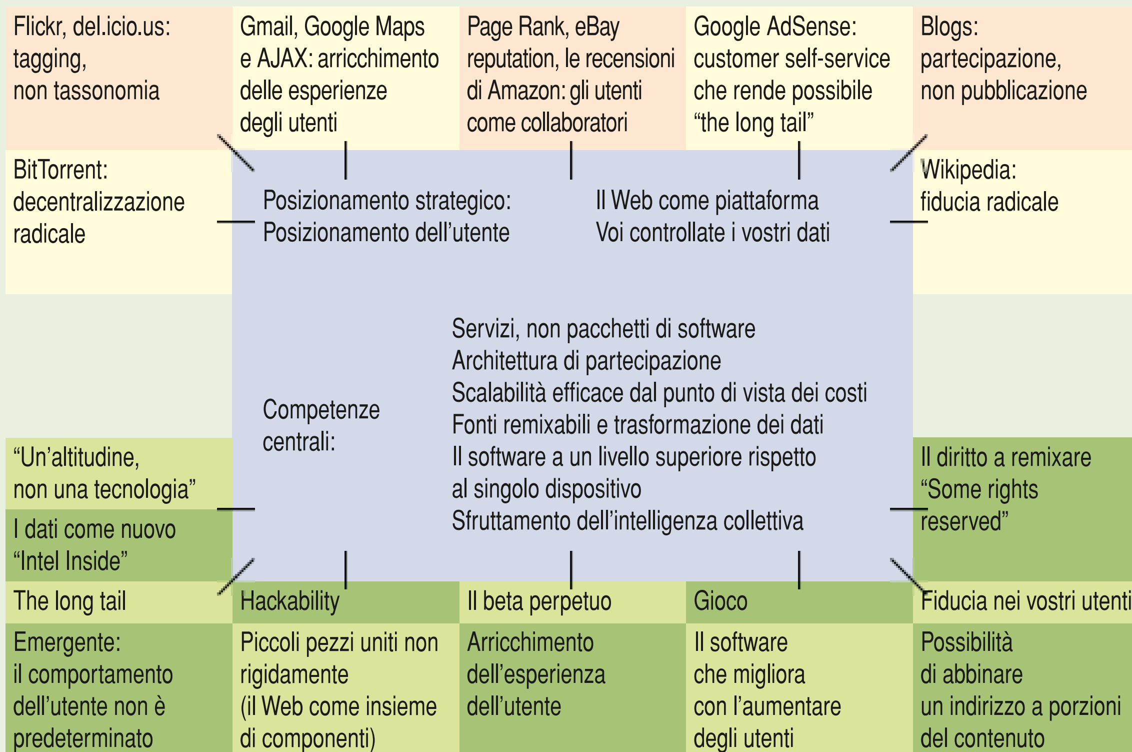
WEB 2.0

Il termine Web 2.0 non si riferisce alla nuova versione di un software bensì all'evoluzione sociale che ha visto protagonista il mondo di Internet negli ultimi anni: tramite il Web 2.0 Internet si configura sempre più come "Social Network" dove la collaborazione assume il ruolo di fattore chiave per l'innovazione e lo sviluppo. L'uso del linguaggio XML in sostituzione dell'HTML rende facilmente fruibili e reinterpretabili le informazioni provenienti da applicazioni differenti. L'idea principale su cui si basa la nuova filosofia è la possibilità di trasferire qualsiasi applicazione comunemente utilizzata dal PC al Web. Dietro queste evoluzioni troviamo tecnologie quali:

- Web Service
- API
- XML
- RSS
- Ajax (Asynchronous JavaScript and XML)

Le applicazioni più diffuse del Web 2.0 sono:

- Blog
- Wiki
- Social network
- Podcasting
- Vodcast



Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Indica la differenza tra router, ISP e host.

a) Router instrada i pacchetti, mentre l'ISP distribuisce le connessioni ai web client e host è sinonimo di computer che ospita fisicamente un sito
b) Router distribuisce le connessioni ai web client, mentre l'ISP instrada i pacchetti e host è sinonimo di computer che ospita fisicamente un sito
c) Router è sinonimo di computer che ospita fisicamente un sito, mentre l'ISP distribuisce le connessioni ai web client e host è il computer che instrada i pacchetti
d) Router instrada i pacchetti, mentre l'ISP distribuisce le connessioni di posta elettronica e host è sinonimo di computer che ospita fisicamente un sito

2 Metti in ordine logico le operazioni seguenti che avvengono durante una comunicazione client-server secondo il protocollo HTTP.

a) .... si apre una connessione TCP tra client e server
b) .... si chiude la connessione
c) .... il server risponde (se possibile, fornendo la risorsa richiesta)
d) .... il browser richiede una risorsa al server http (web server)

3 Metti in ordine logico le seguenti operazioni che avvengono a livello client in una conversazione http.

a) .... la pagina multimediale è caricata sul browser
b) .... il browser apre una connessione TCP con il web server (porta 80)
c) .... il browser analizza l'URL e ne estrae il dominio
d) .... il browser consulta il DNS per ottenere l'indirizzo IP corrispondente al dominio
e) .... una volta stabilita la connessione, il browser e il server usano http per comunicare

4 L'URL ftp:miosito.com indica:

a) che il protocollo usato è ipermediale e che il sito si trova a Roma
b) che il protocollo usato è ipermediale e che il sito si trova in Italia
c) che il protocollo usato è di trasferimento file e che il sito si trova a Roma
d) che il protocollo usato è di trasferimento file e che il sito è di tipo commerciale

5 Indica il protocollo dal livello fisico più basso tra i seguenti protocolli:

a) TCP b) IP c) FTP d) TELNET

6 Indica quale definizione di Internet è corretta.

a) rete multimediale
b) insieme di computer
c) insieme di host
d) rete di reti

7 Quali tra i seguenti sono linguaggi che possono essere inclusi in pagine HTML?

a) VBScript b) php c) Cobol
d) Java e) Pascal f) JavaScript

8 Il protocollo http permette:

Il trasferimento di file
di avere una connessione in tempo reale tra due host remoti
il trasferimento di dati ipertestuali
il trasferimento di file per la posta elettronica

9 Dove è definito il formato dei messaggi http?

a) uni b) rfc c) iso d) osi

10 Da che cosa è composto un messaggio HTTP?

a) da una start-line e da una body
b) da una body e una head e da una response
c) da una start-line, da una head e una body
d) da una start-line, da una head, da una body e da una request

Test vero/falso

1 Una applicazione distribuita è composta da più elementi cooperanti posti sulla stessa macchina. V F
2 Un browser è un software che decodifica pagine scritte in formato HTML. V F
3 Dal browser non è possibile digitare l'indirizzo IP dell'host al quale connettersi. V F
4 HTML, GALEON, LITE e MOZILLA sono browser in commercio. V F
5 Intranet e internet usano i medesimi protocolli. V F
6 Un router è un instradatore di pacchetti. V F
7 WWW è l'insieme delle pagine ipermediali di internet. V F
8 Protocollo di comunicazione è sinonimo di linguaggio. V F
9 I DNS contengono le tabelle di routing. V F
10 HTTP è un protocollo inferiore rispetto a TCP/IP. V F
11 IIS è un browser. V F
12 I messaggi HTTP possono essere divisi nelle categorie request e response. V F
13 Il protocollo File Transfer Protocol viene utilizzato per il trasferimento di file solo in upload. V F
14 I protocolli SMTP e POP vengono utilizzati nel trasferimento dei file. V F
15 Una URL può contenere anche una directory. V F
16 Un web server è una macchina sulla quale gira un programma php. V F
17 Apache e PWS sono entrambi web server di rete. V F

Domande aperte

1 Quali sono le applicazioni più importanti di Internet?

....

2 Cosa significa FTP?

....

3 Quanti tipi di protocolli di comunicazione conosci per i livelli intermedi?

....

4 Spiega con parole tue il significato di browser.

....

....

5 Indica almeno 5 browser diversi.

6 Dai una definizione di router.

7 Quale protocollo tra IP e HTTP è a un livello più basso?

8 Cosa si intende per ipermedialità?

9 Spiega la differenza tra request e response.

10 Indica la dinamica del dialogo tra web client e web server mediante protocollo http.

11 Spiega da quali parti è formato un messaggio http.

12 Se il codice di risposta del server è 202, cosa significa?

13 Se il codice di risposta del server è 404 che cosa significa?

14 Cosa contiene la body della client-request se il metodo scelto è POST?

15 Spiega a cosa serve il protocollo ftp.

16 Spiega il significato di tutte le parti che compongono il seguente indirizzo URL: http://www.piemonte.regione.it/inventari/anagrafica/1997/index.php

LEZIONE 4

TRASFERIMENTO DI FILE: FTP

IN QUESTA LEZIONE IMPAREREMO...

- il client e il server FTP
- le modalità di collegamento FTP
- i comandi FTP

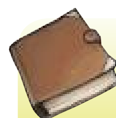
Generalità

È difficile avere un'idea del numero di file che sono oggi "sparsi" nelle memorie degli *host* computer connessi tra loro in Internet, tra programmi, immagini, documenti, suoni, filmati ecc., così come è difficile calcolare proprio il numero degli host presenti sulla rete, numero che aumenta giorno per giorno.
Senza timore di eccedere possiamo affermare che milioni di file sono condivisi in rete e in ogni istante una parte di essi viene scambiata tra calcolatori presenti in Internet, sia tra macchine molto distanti tra di loro che tra macchine direttamente connesse, presenti nello stesso locale.

Il protocollo utilizzato a livello applicativo per trasferire i file è il File Transfer Protocol, comunemente chiamato FTP, che si basa su TCP ed è stato definito nel 1984 nell'RFC 959.

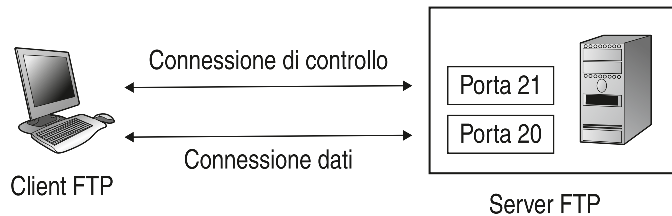
Gli obiettivi principali dell'FTP, descritti nella sua specifica ufficiale, sono:
- promuovere la condivisione di file (programmi o dati);
- incoraggiare l'uso indiretto o implicito di computer remoti;
- risolvere in maniera trasparente incompatibilità tra differenti sistemi di deposito file;
- trasferire dati in maniera affidabile ed efficiente.

A differenza di altri protocolli, FTP per trasportare un file utilizza due canali TCP separati che agiscono in parallelo, cioè una connessione di controllo e una connessione dati;
- la prima connessione è usata per spedire le informazioni di controllo tra client e server, come l'identificazione dell'utente, password, comandi per



OUT OF BAND
Dato che l'FTP usa una connessione di controllo separata, si dice che l'FTP invia le sue informazioni di controllo fuori banda (out-of-band).

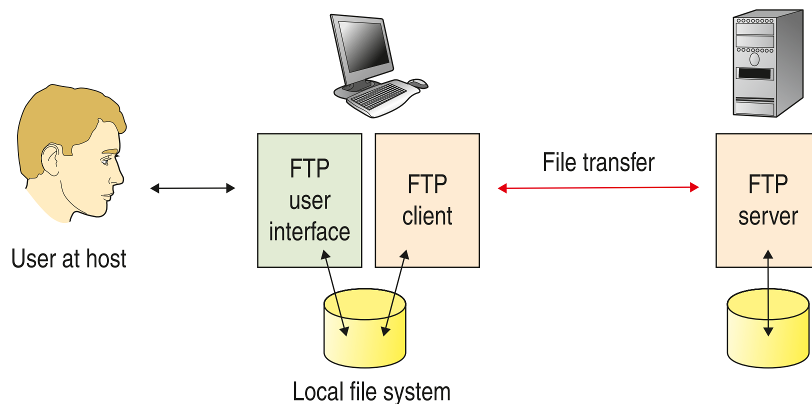
la variazione della directory remota ecc.): viene normalmente chiamata control connection o command channel;
- la seconda connessione è quella effettivamente utilizzata per il trasferimento dei file: viene normalmente chiamata data connection o data channel.



Il server e il client FTP

Il protocollo FTP si riferisce a un modello di tipo client/server dove la macchina host destinata a svolgere la funzione di server ha in esecuzione uno specifico programma che può essere fornito direttamente col sistema operativo oppure istallato in un secondo tempo.

Per poter realizzare la connessione FTP entrambe le macchine devono avere istallato un software FTP, rispettivamente un FTP client e un FTP server.



FTP Server

Esistono in distribuzione parecchi software FTP validi, sia distribuiti gratuitamente con licenza open source, sia a pagamento, come per esempio:
- Filezilla Server;
- Bulletproof Ftp Server;
- Golder Ftp Server;
- Globalscape Secure Ftp Server;
- Cerberus Ftp Server;
- Gene6 Ftp Server;
- Serv-U Ftp Server;
- Wftpd;
- Surgeftp.

Il software FTP server mette a disposizione dei client molteplici opzioni per interagire con l'insieme dei file condivisi nel suo file system, tra cui:
- download/upload di file;
- recupero (resume) di trasferimenti interrotti;

- rimozione e rinomina di file;
- creazione di directory;
- navigazione tra directory.

L'accesso al FTP server viene effettuato mediante un sistema di autenticazione e in base alle credenziali possono essere assegnati determinati privilegi agli utenti registrati per poter operare sul file system: spesso è anche possibile effettuare l'autenticazione "anonima", cioè quella che effettua un client che non specifica alcuna password di accesso: in questo caso, generalmente, gli viene concesso come privilegio quello di poter effettuare la "sola lettura".

FTP Client

Anche la connessione da parte del client avviene mediante un apposito software, generalmente gratuito, capace di connettersi a un server e di caricare file dal proprio computer al server, quindi ◀ uploadare ▶ il contenuto a un indirizzo web, o downloadare file da un server a un particolare indirizzo IP.
Generalmente un FTP client è un programma costituito da due componenti:
- parte di comunicazione: implementa il protocollo FTP;
- interfaccia utente: generalmente di tipo grafico agevola lo svolgimento delle funzioni offerte dal prodotto.



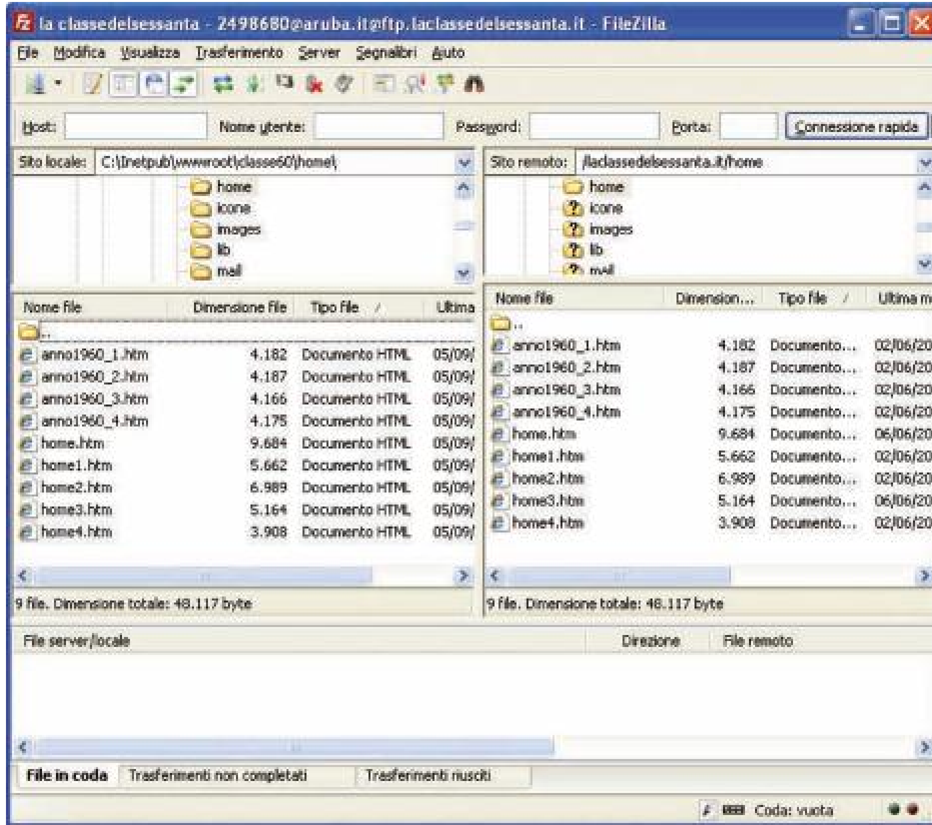
◀ Uplodare Uplodare o uploadare? Sono molteplici i problemi di natura terminologica posti dagli anglicismi tra i quali quelli dovuti ad adattamenti di grafia e di flessione nelle famiglie formate con corradicali, come uploadare/uplodare da upload "caricamento" e downloadare/downlodare da download.
Sarebbe opportuno utilizzare l'italiano "caricare" e "scaricare" e limitare l'uso dell'anglicismo solo ai casi in cui non si dispone di soluzioni alternative. ▶

I diversi programmi si differenziano dalle opzioni offerte all'utente, che vanno da quella di permettere upload multipli, alla scelta della velocità, al transfer da server a server, ai componenti grafici dello stato e dell'avanzamento dei trasferimenti, all'interfaccia grafica accattivante, alla possibilità di interrompere il download quando si vuole e riprenderlo senza file corrotti.

Le funzioni comuni a tutti i programmi sono le seguenti:
- connessione al server remoto;
- trasferimento di un file dal server al client (get);
- trasferimento di un file dal client al server (put);
- lettura dei files presenti nella directory corrente sul server (dir);
- cambiamento di directory corrente sul server (cd);
- disconnessione (bye).

Di seguito è riportato un elenco parziale del principali client FTP:
- Ace FTP freeware;
- Directory uploader;
- Racing turtle FTP;
- Filezilla;
- Cute FTP;
- TRELLIAN;
- WSFTP;
- Fastream netfile FTP;
- Blazeware FTP;
- Smart FTP;
- Cesar FTP.

Alcuni dei prodotti elencati possono fare simultaneamente da server e da client, hanno cioè al loro interno entrambe le funzionalità.



In figura è riportata l'interfaccia grafica di Filezilla, utilizzato in questo esempio per uploadare file su un server web: è immediato individuare il file system locale (a sinistra) e il file system remoto (a destra).
I file vengono caricati/scaricati semplicemente mediante operazioni di trascinamento (◀ **drag and drop** ▶).



◀ **Drag and drop** Drag-and-drop is the action of selecting a virtual object by "grabbing" it and dragging it to a different location or onto another virtual object. ▶

La comunicazione FTP

Analizziamo ora il funzionamento del protocollo FTP seguendo le operazioni necessarie per effettuare il trasferimento di un file dal server al client.
Abbiamo sostanzialmente due possibili situazioni:
1. collegamento normale;
2. collegamento passivo.

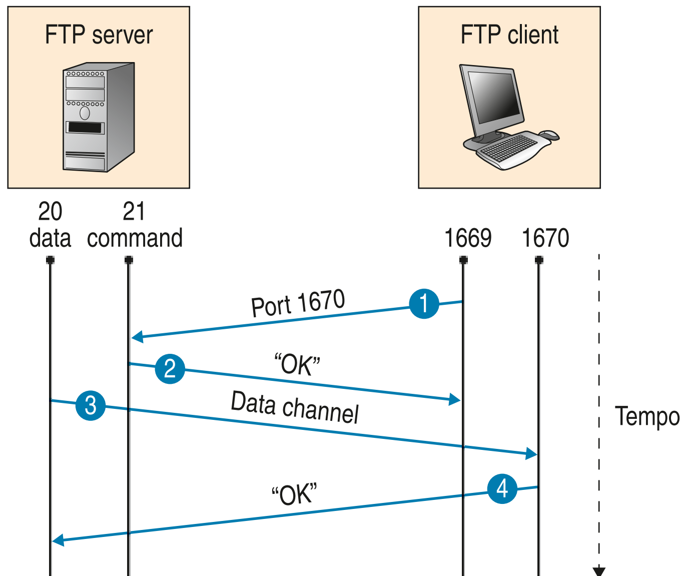
Collegamento normale (*normal-mode*)

Quando un utente avvia una sessione FTP con un server remoto, il lato client dell'FTP come prima cosa apre due porte con numero random superiore a 1023, una per la trasmissione e ricezione dei dati e l'altra per i segnali di controllo (nel nostro esempio utilizziamo rispettivamente le porte 1669 e 1670).

La successiva operazione del client è quella di instaurare una connessione TCP di controllo con il server (host remoto) sulla porta 21 del server, specificando TCP come protocollo di trasporto e il numero di porta dati del client mediante il comando port 1670: il client ottiene dal server l'autorizzazione alla connessione sulla linea di controllo (l'"OK" sulla porta 1669).

Il client dell'FTP invia l'identificazione dell'utente (nome utente) e la password su questa connessione di controllo, nonché gli eventuali comandi per cambiare la directory remota, sempre comunicando sulla connessione di controllo.

Quindi il server apre il data channel verso la porta indicata dal client per iniziare il trasferimento di un file utilizzando la porta 20.
Al termine del trasferimento di un file, il server chiude la connessione sulla linea dati.



Osserviamo che per tutta la durata della sessione la connessione di controllo rimane sempre aperta mentre viene chiusa la linea dati al termine del trasferimento di un file: per ricevere un nuovo file è necessario stabilire una nuova connessione dati, sempre comunque all'interno della stessa sessione:
- la connessione di controllo è persistente;
- la connessione dati non è persistente.

Questo tipo di collegamento tra client e server è denominato "normale" (normal-mode).

Collegamento passivo (*passive mode* o *PASV mode*)

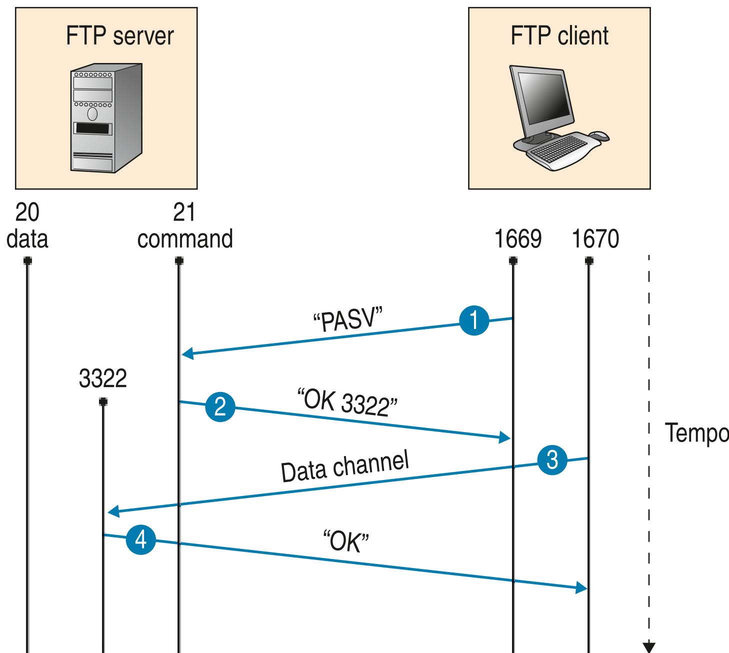
Durante una sessione, il server FTP deve mantenere lo stato dell'utente e associa a una connessione lo specifico account dell'utente e le directory dell'host remoto: queste informazioni di stato devono essere mantenute per ciascuna delle sessioni dell'utente in corso e comportano una limitazione sul numero totale delle sessioni che l'FTP può mantenere simultaneamente.
Per cercare di liberare il più possibile la banda, se non viene effettuata nessuna operazione per un certo tempo, viene interrotta automaticamente la sessione FTP: questo comporta che se l'utente vuole compiere altri trasferimenti deve riaprire nuovamente la connessione di controllo.

Esiste anche un metodo alternativo che viene supportato da quasi tutti i server FTP: la modalità denominata "passiva" (passive mode o PASV mode), che consente ai client di aprire sia la connessione di controllo, sia quella per i dati.

Come per la connessione normale, il client nella connessione FTP passiva alloca due porte con numero maggiore di 1023 (nel nostro esempio sempre le porte 1669 e 1670), ma non viene indicato al server il numero di porta riservata ai dati: è il server a comunicare al client il numero di porta che lui riserva per il trasferimento dei dati (per esempio la porta 3322), e quindi non è più la porta numero 21 di default, come nel caso normale.

Al posto del numero di porta data, il client comunica al server il comando PASV per indicare l'uso della modalità passiva.

La risposta del server comprende anche il suo numero di porta data 3322 in modo che il client attivi la connessione anche su di essa, inviando un comando di richiesta di apertura per il canale DATA (Data Channel 1670-3322).



Questo meccanismo rende la trasmissione più sicura di quella normale in quanto è sempre il client FTP ad attivare una connessione, che sia quella di controllo oppure quella per i dati: il client non fornisce al server la porta su cui accetta la connessione dati evitando così che dei malintenzionati possano sfruttare questa porta "aperta" per intrufolarsi nel sistema e sferrare un attacco "hacker".

Comandi comuni

Il client invia i comandi al server FTP come normale testo ASCI, generalmente composti da due elementi: `nome_comando <parametro>`.

ESEMPIO 8

Sono riportati di seguito a titolo di esempio un gruppo di comandi:

- `USER <nome utente>` identifica l'utente;
- `PASS <password>` autentica l'utente;
- `LIST` elenca i file della directory corrente;
- `RETR <filename>` recupera (get) un file dalla directory corrente;
- `STOR <filename>` memorizza (put) un file nell'host remoto;
- `CWD <directory>` cambia directory corrente;
- `QUIT` esegue la disconnessione.

Anche le risposte che il server FTP invia al client sono costituite da testo ASCII, in particolare hanno una stringa composta da tre caratteri (XYZ) che assumono il seguente significato:

- il primo carattere X:
  - 1 = l'azione richiesta è stata iniziata;
  - 2 = l'azione richiesta è stata completata con successo;
  - 3 = il comando è stato accettato ma è in sospeso in quanto sono necessarie altre informazioni quali USER, PASS;
  - 4 = il comando non è stato accettato dato che si è verificata una situazione di errore provvisoria;
  - 5 = il comando non è stato accettato dato che si è verificato un errore irrecuperabile;

- il secondo carattere Y:
  0 = presenza di un errore di sintassi;
  2 = dà informazioni sullo stato della connessione;
  3 = dà informazioni sullo stato della autenticazione dell'utente;
  5 = indica lo stato del file system del server;
- il terzo carattere Z non è rigidamente codificato come i primi due ma è in funzione del valore che essi assumono.

ESEMPIO 9

Vediamo alcuni esempi di codici di stato con l'espressione di ritorno

331 Username OK, password required	nome utente accettato, si attende la password
125 Data connection already open; transfer starting	connessione dati aperta, si inizia il trasf. del file
425 Can't open data connection	errore: non si riesce ad aprire una connessione dati (impossibile trasferire un file)
452 Error writing file errore	non è possibile scrivere il file

È presente una eccezione: il codice 220 indica accettazione della connessione e insieme a esso viene comunicata la chiave pubblica del server.

ESEMPIO 10 *Connessione FTP client-server*

Vediamo come avviene una normale procedura di connessione tra un client (C) e un server (S) utilizzando i comandi sopra elencati:

C) il client esegue una connessione al server collegandosi alla porta TCP 21

S) il server risponde

C) il client invia i dati necessari per identificare l'utente partendo da:
```
USER <nome utente>
```

S) il server risponde positivamente 331 richiedendo la password
331 username ok, password required

C) il client invia ora la password:
```
PASS <password>
```

S) il server risponde che l'utente è correttamente autenticato, oppure nega l'accesso

C) se l'utente è autenticato, il client può dare i vari comandi previsti dal protocollo, per esempio con uno schema come il seguente:

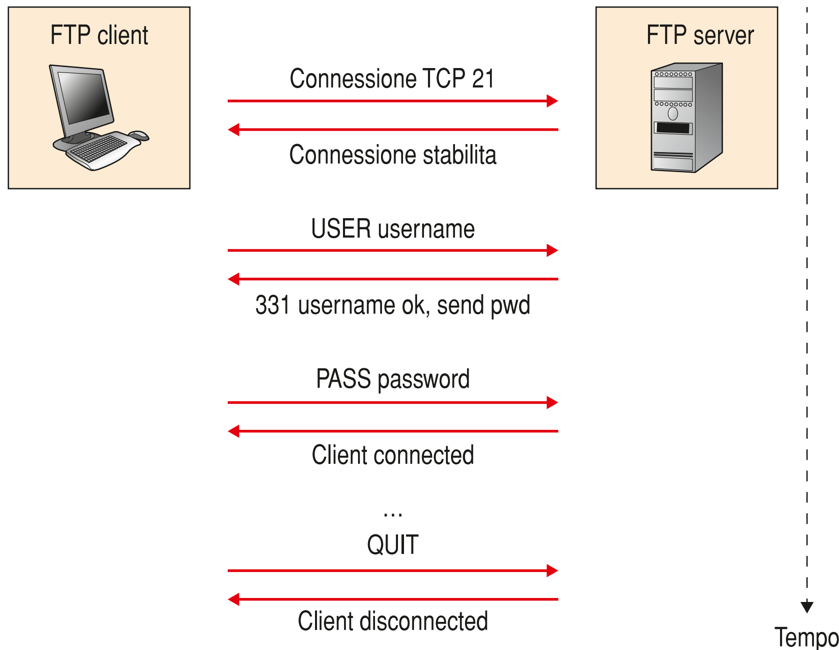
C) il client invia per esempio il comando
```
RETR <filename>
```

S) il server risponde con un codice di stato e una descrizione
125 Data connection already open; transfer starting

Il trasferimento di file è uno scambio di dati, mentre il colloquio tra client e server è uno scambio di comandi.

C) l'utente si disconnette
```
QUIT
```

Schematicamente abbiamo la situazione rappresentata in figura ▼



FTPS

Sempre in termini di sicurezza, è doveroso ricordare che il protocollo FTP non prevede alcuna cifratura per i dati scambiati tra client e server e tra questi vengono scambiati nomi utenti, password, comandi, codici di risposta e file trasferiti che possono essere agevolmente "sniffati" da malintenzionati.

Per ovviare a questo problema è stata definita una nuova specifica, la RFC 4217, che aggiunge al protocollo FTP originale un layer di cifratura ◀ **SSL** ▶ più una nuova serie di comandi e codici di risposta: tale protocollo prende il nome di FTPS.

> Tutti i trasferimenti FTPS sono cifrati e l'algoritmo di cifratura è negoziato con il server: se viene impostato la finestra del trasferimento mostrerà un simbolo di un "lucchetto" a indicare il trasferimento sicuro.



◀ **SSL** Short for Secure Sockets Layer, a protocol for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data – a public key known to everyone and a private or secret key known only to the recipient of the message. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with https: instead of http. ▶

Non bisogna confondere il protocollo FTPS dal servizio SFTP: FTPS è un protocollo definito a livello "ufficiale" che cifra solamente la sessione e non tutti i dati che vengono scambiati, al contrario di SFTP che non è altro che una sessione cifrata (SSH) che utilizza comandi FTP-like per trasferire i files sensibili che, quindi, sono intermanete cifrati.



Zoom su...

TIPI DI CONNESSIONI FTPS

Sono possibili tre diversi tipi di connessione FTPS:

- **A** solo controllo: crea una connessione sicura FTP (FTPS) con un layer SSL al di sotto del protocollo standard FTP cifrando SOLO il canale di controllo.
- **B** controllo + dati: crea una connessione sicura FTP (FTPS) con un layer SSL al di sotto del protocollo standard FTP cifrando sia il canale di controllo sia il canale dati.
- **C** implicito: crea una connessione sicura FTP (FTPS) con un layer SSL al di sotto del protocollo standard FTP. Questo è un vecchio metodo FTPS e generalmente non è raccomandato, ma alcuni server potrebbero ancora supportarlo.

Conclusioni

Dato che il paradigma adottato nel protocollo FTP è del tipo client/server non è possibile per un client stabilire una connessione FTP verso una qualsiasi macchina: è necessario che il server sia stato opportunamente configurato per accettare la connessione e operare il trasferimento dei file.
Se si tenta di stabilire una connessione verso una macchina non è abilitata a offrire il servizio FTP la sessione fallisce e nessun trasferimento risulta possibile: non tutti gli host sono configurati di default per accettare connessioni di tipo FTP.

Generalmente le macchine server Linux o Windows Server hanno il servizio FTP già attivo, senza che esso debba essere esplicitamente installato, mentre le macchine client dotate di sistema Windows non accettano automaticamente connessioni FTP ma devono essere opportunamente configurate.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Il protocollo FTP è stato definito:

a) nel 1984 nell'RFC 659
b) nel 1984 nell'RFC 959
c) nel 1994 nell'RFC 659
d) nel 1994 nell'RFC 959

2 Gli obiettivi principali dell'FTP sono (indica quello inesatto):

a) promuovere la condivisione di file programmi
b) promuovere la condivisione di file dati
c) realizzare l'uso indiretto o implicito di computer remoti
d) risolvere in maniera trasparente incompatibilità tra differenti sistemi di deposito file
e) trasferire dati in maniera affidabile ed efficiente

3 Indica tra i nomi dei canali seguenti quelli che sono tra loro sinonimi:

a) command connection o control channel
b) control connection o command channel
c) data connection o data channel
d) data connection o file channel

4 Il software FTP server mette a disposizione le seguenti opzioni (indica quella errata):

a) download di file
b) upload di file
c) cifratura dei file
d) recupero di trasferimenti interrotti
e) rimozione e rinomina di file
f) creazione di directory
g) navigazione tra directory

5 Il trasferimento di un file dal server al client può essere fatto con:

a) collegamento normale
b) collegamento diretto
c) collegamento attivo
d) collegamento passivo

6 Durante una sessione FTP:

a) la connessione di controllo è persistente
b) la connessione di controllo non è persistente
c) la connessione dati è persistente
d) la connessione dati non è persistente

Test vero/falso

1 Il protocollo FTP è a livello di trasporto. V F
2 FTP per trasportare un file utilizza due canali HTTP separati che agiscono in parallelo. V F
3 Un canale si occupa di trasferire i comandi e l'altro i dati. V F
4 FTP invia le sue informazioni di controllo *fuori fase*. V F
5 Il protocollo FTP si riferisce a un modello di tipo client/server. V F
6 Un FTP client è un programma costituito da due componenti: *parte di comunicazione e interfaccia utente*. V F
7 Un server FTP utilizza le porte 20 e 21. V F
8 Il PASV mode consente ai client di aprire sia la connessione di controllo sia quella dati. V F
9 Il PASV mode utilizza la porta 21 per trasmettere i dati. V F
10 Il server comunica al client al posto del numero di *porta data* il comando PASV. V F

Domande aperte

1 Prova a descrivere il protocollo FTP.

2 Perché esiste il modo FTP passivo? Come funziona?

3 Il protocollo FTP usa connessioni persistenti? Descrivi il funzionamento e i comandi che ricordi.

4 Quali sono i problemi connessi con la sicurezza?

5 Quali sono le principali differenze tra il protocollo FTPS e il servizio SFTP?

LEZIONE 5

POSTA ELETTRONICA IN INTERNET: SMTP, POP E IMAP

IN QUESTA LEZIONE IMPAREREMO...

- il funzionamento della posta elettronica
- i meccanismi dei protocolli SMTP, POP e IMAP

Generalità

Il successo di Internet è stato anche determinato dal servizio di posta elettronica o più correttamente email (dall'inglese «electronic mail»), che senza dubbio è l'applicazione più utilizzata in rete, sia da privati, per comunicare con amici al posto del tradizionale documento cartaceo, sia da aziende, per le normali transazioni commerciali e per inviare ai possibili clienti quantità "sproporzionate" di messaggi commerciali (◀ spam ▶) dato che il loro costo è praticamente nullo.



◀ Spam La parola spam (spiced ham) era usata negli Stati Uniti d'America per indicare un tipo particolare di carne in scatola, fornita all'esercito americano, di pessima fama. Nel gergo di Internet spam indica l'insieme dei messaggi email indesiderati, soprattutto di tipo pubblicitario. ▶

Gli indirizzi di posta elettronica, in Internet, hanno la forma:

nomeutente@dominio

dove
- nomeutente è la stringa che identifica l'interlocutore (mittente o destinatario) che è univoca per dominio: è un nome scelto dall'utente o dall'amministratore del server, che identifica in maniera univoca un utente;
- dominio è rappresentato dal nome del provider e dalla nazionalità oppure da un indirizzo IP.



◀ @ Il carattere chiocciola (@, in inglese "at") è usato nella posta elettronica per indicare "presso". Nasce intorno al VI-VII secolo, più tardi viene usato in Inghilterra per scopi commerciali e stava a indicare "al prezzo di". Per esempio, in fattura si scriveva: *five barrels @ 200 pence each*.
Quando nel 1972 venne creata la rete ARPANET (antenata di Internet) venne usata per la prima volta la chiocciola per separare il nome utente dal server negli indirizzi email. ▶

ESEMPIO 11

L'indirizzo seguente possiede come nome utente mario rossi, come dominio il provider Tiscali e come nazionalità it (Italia).

mariorossi@tiscali.it

L'indirizzo email può contenere qualsiasi carattere alfabetico e numerico (escluse le accentate) e alcuni simboli come l'underscore (_) e il punto (.).

Dobbiamo sottolineare come l'indirizzo di posta elettronica non sia associato a una persona, ma a una casella postale elettronica; ogni utente può possedere diverse caselle come è anche possibile associare nomi utente diversi (alias) alla stessa casella di posta.
È inoltre doveroso fare una precisazione differenziando la posta elettronica in funzione delle modalità di accesso a essa, che può essere di due tipi:
- POP mail;
- Web mail.

La POP mail permette di leggere e inviare i messaggi di posta elettronica dal computer nel quale è installato il programma di posta elettronica preferito, genericamente chiamato ◀ client di posta ▶. Ne esistono molti in commercio tra i quali, per esempio, Outlook Express, Eudora, IncrediMail, Mozilla Thunderbird ecc.



◀ Client di posta Viene anche chiamato Mail User Agent (MUA e verrà descritto in seguito) e si tratta di un programma molto diffuso che consente di comporre, inviare, ricevere e organizzare i messaggi di posta elettronica da e verso il server di posta elettronica di un ISP (*Internet Service Provider*) al quale siamo registrati. ▶

La Web mail invece consente di accedere alla casella di posta elettronica solo attraverso la connessione a un sito dedicato a questo scopo come, per esempio, Hotmail, GMail ecc. La seguente tabella ne illustra le principali caratteristiche.

Web mail	POP mail
Gratuita	Spesso a pagamento
Accesso da qualunque computer con connessione a Internet	Accesso solo da computer nel quale è installato il programma di posta elettronica (client)
Necessità di ricordare la password a ogni accesso	Password memorizzata solo durante la creazione dell'account
Metodo poco sicuro, le password possono essere intercettate	Metodo sicuro in quanto le password sono memorizzate nel computer client
Gli allegati sono consentiti solo se sono di dimensioni limitate	Gli allegati possono essere di dimensioni molto elevate
Rischio di ricevere molte email indesiderate (spam)	Rischio spam ridotto

Invio e ricezione di posta elettronica

Dopo aver scritto una email utilizzando un qualunque programma di posta con l'indicazione dell'indirizzo del destinatario, provvediamo a inoltrarla sulla rete.

> Analogamente al sistema postale cartaceo, non è necessario che il ricevente sia connesso alla rete e abbia in funzione anch'esso un programma di posta elettronica: al momento della trasmissione della posta da parte del mittente il destinatario potrebbe essere impegnato oppure avere il computer spento.

La email non raggiunge quindi immediatamente il computer del destinatario, ma viene depositata in una memoria, come nel "fermo posta" degli uffici postali, in attesa che qualcuno la "passi a ritirare."

Vediamo dettagliatamente come avviene il trasferimento dei messaggi di posta elettronica tra gli intermediari, così come è stato definito in RFC 822: viene utilizzato il protocollo Simple Mail Transfer Protocol (SMTP).

La realizzazione della posta elettronica viene implementata mediante due sottosistemi:
- Mail User Agent (MUA);
- Mail Transport Agent (MTA).

MUA

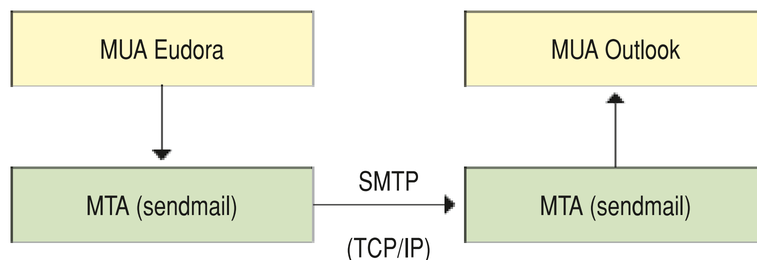
Il MUA, come già detto precedentemente, è un programma di gestione di posta che viene mandato in esecuzione sul client e mediante una interfaccia grafica user friendly offre all'utente le funzionalità che permettono l'inserimento, la composizione, la ricezione e la lettura dei messaggi.

> Come vedremo, i messaggi devono essere strutturati secondo una sintassi di composizione definita nel RFC 822 e nel MIME.

Per spedire i messaggi MUA utilizza il protocollo SMTP e li consegna a un programma MTA che si occupa della trasmissione al MUA destinatario.

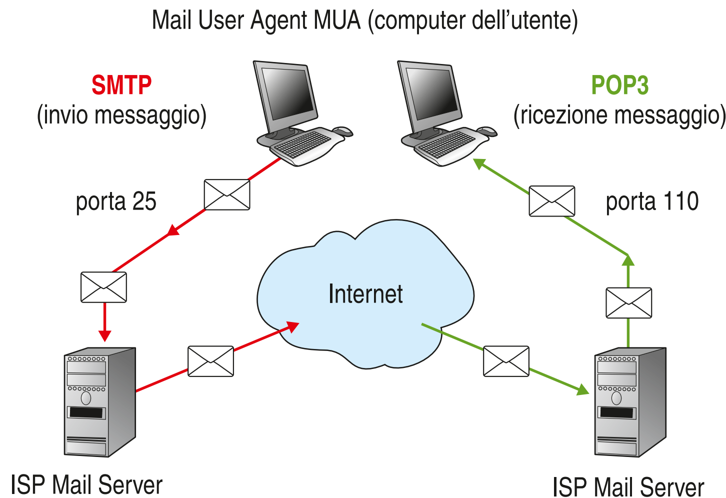
MTA

L'MTA si occupa della ricezione di tutti i messaggi e del loro recapito e ha quindi la funzione di ponte tra due MUA: è quello che noi indichiamo come mailserver.
Può essere paragonato a una centralina telefonica dove vengono smistate le chiamate in ingresso e in uscita, e offre i seguenti servizi:
- server SMTP (porta 25): gestisce la spedizione e la ricezione dei messaggi tra server SMTP;
- server POP3 (porta 110): gestisce la spedizione dei messaggi al client;
- server IMAP4 (porta 143): permette la gestione dei messaggi sul server dal client.



Il MUA è quindi una applicazione in esecuzione su un server dell'ISP dell'utente: questo compone e invia il messaggio verso il proprio ISP che si occupa della ricezione di tutti i messaggi e del loro recapito al mail server dell'IPS di destinazione, raggiungibile dal destinatario del messaggio.

Lo schema completo di trasmissione e ricezione è riportato nella seguente figura.



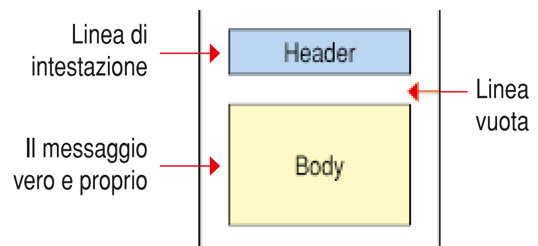
I mail server hanno al loro interno la mailbox "fermo posta" degli utenti contenente i messaggi in entrata che non sono ancora stati letti e scaricati sui rispettivi client nel caso di gestione in modalità POP mail: nel caso di Web mail al MTA vengono aggiunte anche tutte le funzionalità del sottosistema MUA dato che l'utente effettua tutte le operazioni direttamente sul server.
Inoltre, per entrambe le situazioni, i mail server hanno memorizzato la coda dei messaggi in uscita contenente i messaggi non ancora recapitati.
L'invio del messaggio da parte del client mail avviene utilizzando il protocollo SMTP mentre la ricezione viene effettuata con il protocollo POP3.

■ Il protocollo SMTP

Il primo documento che descrive il protocollo per la rappresentazione dei documenti di posta elettronica è l'RCF 822, del 1982; in esso viene specificato il formato per i messaggi di posta ma limitandosi ai messaggi di testo, cioè esclusivamente a simboli (caratteri) di 7 bit in formato ASCII, senza fare alcun riferimento ad altri contenuti come per esempio le immagini.

Il formato del messaggio

L'RCF 822 è una stringa di testo costituita da un header e da un body separati da una linea vuota. ▶
L'header contiene le informazioni per il trasporto:



To:	lista di destinatari
From:	mittente
Cc:	lista di destinatari per conoscenza
Bcc:	lista nascosta di destinatari per conoscenza
Date:	data di spedizione
Reply-to:	indirizzo diverso dal mittente
Subject:	titolo del messaggio

Il body, che è il “messaggio vero e proprio”, è costituito da soli caratteri ASCII: nel caso ci fosse bisogno di trasmettere un messaggio costituito non solo da testo ASCII, deve prima essere convertito in questo formato per poter poi essere inviato in rete; inoltre i caratteri ASCII trasportati dal SMTP sono a 7 bit e quindi sono solo i primi 128 mentre all’interno di un file binario possono essere contenuti tutti e 256 caratteri possibili.

Per superare le limitazioni contenute in RFC 822 nel giugno 1992 è stato presentato un nuovo documento, l’RCF 1341, in cui viene descritto lo standard ◀ **MIME** ▶, Multipurpose Internet Mail Extension, nel quale viene specificato come definire il formato sia di messaggi testuali (ASCII e non) sia di messaggi multimediali (cioè contenenti video, suono, immagini ecc.).

Con lo standard MIME è possibile inserire in un qualsiasi messaggio di email, oltre al testo, anche files contenenti immagini, segnali audio e video; il software di gestione della posta si limita a “trasferire” il file al destinatario senza analizzarne il contenuto, che sarà interpretato e decodificato dall’utente che lo utilizzerà in base alle specifiche di tipo inserite nel messaggio stesso.

◀ **MIME** means Multipurpose Internet Mail Extensions, and refers to an official Internet standard that specifies how messages must be formatted so that they can be exchanged between different email systems. MIME is a very flexible format, permitting one to include virtually any type of file or document in an email message. Specifically, MIME messages can contain text, images, audio, video, or other application-specific data. ▶

I meccanismi introdotti in MIME permettono di risolvere i problemi di RFC 822 mantenendo comunque la compatibilità con i documenti scritti secondo il vecchio standard.

Nella testata di un documento MIME si trovano i seguenti campi:
- MIME version;
- Content-Transfer-Encoding;
- Content-Type;
- Content-ID;
- Content-Description.

MIME version: identifica la versione dello standard MIME usato nel messaggio.

Content-Transfer-Encoding: specifica la modalità con cui sono codificati i dati annessi come “accessorio” ai dati principali (allegati); in questo campo viene specificato qual è la relazione tra i dati nella loro forma originale e il formato con cui vengono trasmessi.
I valori di Content-Transfer-Encoding sono: 7bit, 8bit, binary, quoted-printable, base64, x-token.
Il loro significato è questo:
- 7bit, 8bit, binary: questi valori indicano che il messaggio trasmesso non ha subito nessuna codifica aggiuntiva fornendo l’indicazione sul tipo di dati contenuti nel messaggio stesso:
  - 7bit indica che i caratteri sono codificati con 7 bit ASCII;
  - 8bit indica che ogni carattere deve essere interpretato in base a 8 bit e quindi possono essere presenti caratteri non appartenenti al set ASCII;
  - binary indica che il contenuto del messaggio è in formato binario (un’immagine, un file audio ecc.);
- quoted-printable: questo valore indica che il messaggio ora è in formato ASCII ed è stato così trasformato in modo che non subisca altre trasformazioni da parte dei vari sistemi che è costretto ad attraversare prima di giungere a destinazione;
- base64: questo valore indica che i dati sono stati codificati in base64 in modo che ogni carattere venga rappresentato con sei bit;
- x-token: viene usato per specificare uno schema di codifica non standard, definito e personalizzato da chi trasmette il messaggio.



Zoom su...

BASE64

L'operazione di codifica in base64 consiste nel suddividere la sequenza dei bit in ingresso in gruppi di 24 bit; ogni gruppo di 24 bit viene diviso in quattro gruppi di sei bit e a ognuno dei quali si associa il corrispondente carattere ASCII appartenente al sottogruppo specificato.

Content-Type: è strutturato in Content-Type: tipo/sottotipo;[parametro] e indica il sottotipo dei dati contenuti nel messaggio in modo che il software che riceve il messaggio possa immediatamente interpretare i dati che riceve.

Content-ID: identifica il messaggio in modo univoco (opzionale).

Content-Description: è un campo opzionale che descrive il contenuto del messaggio.

Successivamente alla RCF 1341, l'RFC 1847 ha specificato il Secure/MIME (l'S/MIME) che implementa i servizi di sicurezza con la possibilità di inviare messaggi corredati di firma digitale, di crittografia o di autenticazione (viene descritta nel volume 3 del corso di Sistemi e Reti).

ESEMPIO 12

A titolo di esempio riportiamo un header e un body di un messaggio scambiato tra due utenti di info.ciro.it

Received:	from 191.105.104.2 (adamo.info.ciro.it [193.205.204.6]) by eden.info.ciro.it (8.9.3/8.9.3) with SMTP id QAB12345 for <abele@info.ciro.it >; Thu, 11 Jan 2013 12:37:21 +0100 (MET)
Message-Id	<20221001669. QAB12345@ eden.info.ciro.it >
To:	abele@info.ciro.it
From:	caino@info.ciro.it
Subject:	Appuntamento importante
MIME-version:	1.0
Date:	Thu, 11 Jan 2013 12:37:21 MET
X-Mailer:	Endymion MailMan Standard Edition v3.0.22
Content-Type:	text
X-UIDL:	d16f7ac9bfcd49467c84888f8440b62b
Status:	RO

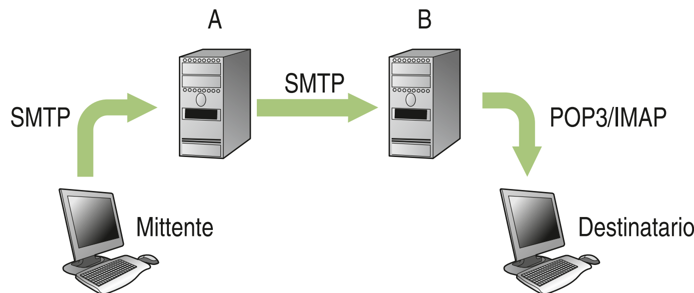
Sono appena arrivato in ufficio: ci vediamo stasera al circolo per il torneo di bridge.

Trasferimento SMTP

Il protocollo SMTP usa il protocollo TCP (porta 25) per consegnare in modo affidabile messaggi dal client al server (l'utilizzo della porta 25 è indicata dalla RCF 1060: "smtp 25/tcp mail").

Prima di vedere il funzionamento di una conversazione SMTP, è necessario premettere che lo standard SMTP è un protocollo "furbo" che per esempio verifica l'esistenza del destinatario prima di inoltrare il messaggio in modo da evitare trasferimenti inutili di dati e che in caso di destinatari multipli cerca di evitare di trasmettere più volte lo stesso messaggio.

L'invio di un messaggio dal mittente al destinatario è composto da tre fasi che coinvolgono i rispettivi server di posta del mittente (A) e del destinatario (B):



1. Il programma di posta elettronica usato dall'utente invia il messaggio al proprio server (A) usando il protocollo SMTP.
2. Il server trasferisce il messaggio al server del destinatario (B) utilizzando lo stesso protocollo:
   - A, sulla base dell'indirizzo email del destinatario, identifica il server B e apre una connessione;
   - B identifica il nodo di rete da cui proviene la connessione (cioè il suo indirizzo IP) e accetta la connessione; memorizza inoltre tale identificazione come parte iniziale del messaggio da ricevere;
   - A comunica l'username del destinatario;
   - B verifica la validità dell'indirizzo e autorizza la trasmissione del messaggio;
   - A invia il messaggio e chiude la trasmissione;
   - B memorizza il messaggio in attesa che il reale destinatario si colleghi e ritiri il messaggio utilizzando un apposito protocollo (solitamente POP3 o IMAP).
3. Il destinatario preleva il messaggio dal proprio server.

Mittente e destinatario si scambiano stringhe di caratteri chiamate comandi: i comandi sono codificati come stringhe ASCII e comprendono un comando in formato testo, un numero di 3 cifre ("reply number") oppure entrambi.

I principali comandi sono:
- helo (Hello): serve per identificare il client e l'argomento contiene l'host name dello stesso client;
- mail from: serve per inizializzare la vera e propria transazione e vuole indicato il mittente;
- rcpt to (recipient): è usato per identificare il destinatario della email e viene ripetuto nel caso di più destinatari contemporanei;
- data: di seguito al comando viene scritto il body della email che termina con un punto (sequenza "<CRLF>.<CRLF>"): se il processo è andato a buon fine il server risponderà con un OK, se invece fallisce si spedirà un'email di reply;
- send: questo comando serve per spedire una email a uno o più terminali ed è seguito dalla email del mittente per comunicargli l'esito;
- soml (send or mail): serve per spedire un messaggio a un terminale oppure alla sua mailbox: se il terminale è attivo riceverà il messaggio, se no verrà recapitato nella sua mailbox;
- saml (send and mail): spedisce il messaggio sia al terminale che alla sua mailbox;

- rset (reset): serve per abortire la transazione email;
- vrfy (verify): permette di effettuare la verifica della email di un utente;
- help: restituisce la lista degli aiuti forniti dal server;
- quit: è il comando di fine della transazione;
- turn: è un particolare comando che serve per invertire i ruoli tra il server e il client.

Nella seguente tabella sono riportati i codici dei “reply number” e il loro significato.

211	System status, or system help reply
214	Messaggio di Help
220	<domain> Servizio pronto
221	<domain> Servizio ha chiuso il canale
250	Richiesta di azione completata, OK.
251	Utente non locale; si spedirà a <forward-path>
354	Inizia l’input dei dati; finisce con <CRLF>.<CRLF>
421	Servizio non avviabile
450	Richiesta di azione non avviabile [E.g., mailbox piena]
451	Richiesta di azione abortita: errore locale durante il processo
452	Richiesta di azione abortita: spazio di sistema insufficiente
500	Errore di sintassi, comando non riconosciuto
501	Errore di sintassi nei parametri o negli argomenti
502	Comando non implementato
503	Cattiva sequenza del comando
504	Parametri del comando non implementati
550	Richiesta di azione non presa: mailbox inaccessibile
551	Utente non locale; per favore prova <forward-path>
552	Richiesta di azione abortita: si eccede l’allocazione di spazio
553	Richiesta di azione non presa: nome della mailbox non permesso
554	Transazione fallita

ESEMPIO 13

Un esempio della conversazione “di comandi” è riportato di seguito, dove possiamo individuare i comandi SMTP e i Reply number tipici del protocollo.

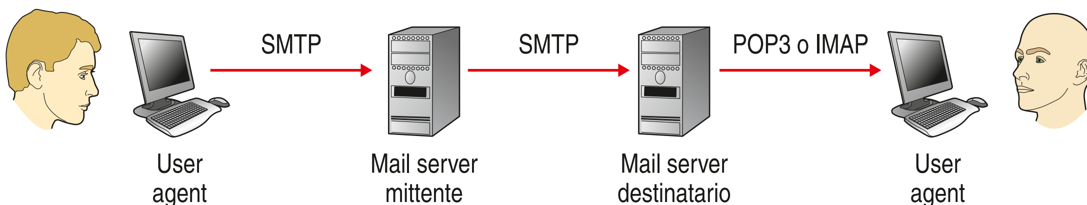
```
S: 220 <ready>                 // server sistemi.edu pronto
C: HELO vacanze.fr
S: 250 <ok>                    // benvenuto vacanze.fr
C: MAIL FROM: <eva@vacanze.fr>   // si precisa l'indirizzo del mittente
S: 250 <ok>                    // pronto a ricevere dati da eva@vacanze.fr
C: RCPT TO: <adamo@sistemi.edu>  // si precisa l'indirizzo del destinatario
S: 250 <ok>                    // pronto a ricevere dati
C: DATA
S: 354                         // iniziano dati che terminano con
                               // <CR><LF>.<CR><LF>
```

```
C: Temo che le vacanze siano finite …
C: Bisogna proprio riprendere a studiare sistemi?
C: <CR><LF>.<CR><LF>
S: 250 <ok>                    // messaggio accettato per la consegna
C: QUIT
S: 221 sistemi.edu             // chiusura connessione
```

Prelievo della posta: Post Office Protocol (POP3)

Fino a ora abbiamo visto il trasferimento dei messaggi tra i vari mail server: ora analizziamo come un utente possa, in un momento qualsiasi, accedere alla propria casella di posta elettronica per leggere i propri messaggi scaricandoli sul disco del proprio PC locale.
Questa operazione viene effettuata mediante il protocollo Post Office Protocol (POP) che permette a un client di posta di accedere al server ed effettuare il trasferimento dei messaggi dalla propria mailbox ("fermoposta") al proprio PC, nel programma client di posta che è installato su di esso.

> Del protocollo POP esistono tre versioni diverse, ma l'unica di fatto utilizzata è la versione 3 (POP3) definito in RFC 1939.



POP3 è un sistema molto semplice che all'avvio del client di posta effettua il log-on al mail server del destinatario e scarica tutti i messaggi a esso pervenuti dai mail server mittenti dall'ultima connessione stabilita: contemporaneamente ai messaggi in arrivo scarica anche gli eventuali attachment a essi collegati.

> Il protocollo POP è anch'esso un comune protocollo client/server dove lo user agent ha ancora il ruolo di client POP e il mail server ha il ruolo di server POP.

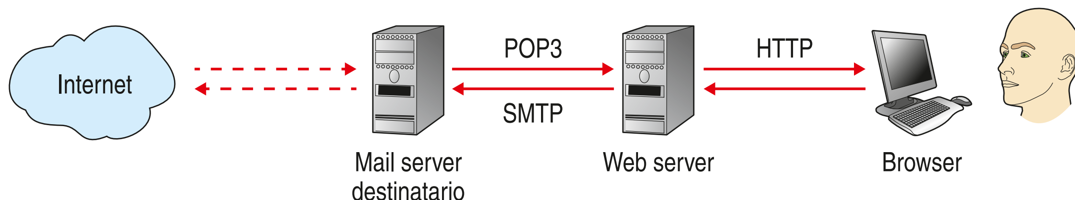
La conversazione POP3, che utilizza la porta 100, avviene in tre fasi.
- autorizzazione (AUTHORIZATION): il client si identifica e il server verifica che abbia le dovute autorizzazioni per accedere alla casella postale;
- transazione (TRANSACTION): è la fase nella quale la posta viene effettivamente scaricata e al termine della stessa il client lancia il comando QUIT;
- aggiornamento (UPDATE): in questa fase il server elimina dalla casella postale tutti i messaggi scaricati e chiude la connessione.

I principali comandi POP3 sono i seguenti:
- user [name]: per effettuare la richiesta d'accesso alla casella postale, dove come parametro viene fornito il nome (a esso segue il comando PASS);
- pass[psw]: parola d'ordine per effettuare la richiesta d'accesso;
- quit: richiesta di fine lavoro;
- stat: comando utilizzato per effettuare la richiesta d'informazioni sullo stato della casella postale;

- list: serve per effettuare la richiesta di informazioni su uno o su tutti i messaggi in casella;
- retr n: richiesta di spedire il messaggio specificato;
- dele n: richiesta di marcare il messaggio da eliminare per poi cancellarlo nella fase di aggiornamento;
- noop: mantiene aperta la connessione anche in caso di non operatività;
- rset: richiede di eliminare tutti i marchi di cancellazione assegnati a seguito di comandi DELE subito, senza attendere la fase di aggiornamento.

L'accesso alla posta via WEB (Web mail)

La seconda modalità di consultazione della posta è quella che viene offerta direttamente dai siti Web che offrono l'accesso alle proprie caselle di posta (Hotmail, GMail, libero, inwind, tin, Yahoo! ecc.) attraverso la connessione a un sito dedicato a questo scopo direttamente con il browser: in questo caso la posta non può essere scaricata sul client ma viene consultata e scritta direttamente in rete sul Web server.



■ Protocollo IMAP

Un protocollo alternativo al POP3, ancora poco diffuso nonostante sia molto più evoluto, è l'Internet Message Access Protocol (IMAP), particolarmente utile per gli utenti "remoti o mobili".

> Il principale vantaggio di IMAP è quello di permettere l'accesso alla posta sia on line che off line e di consentire alcune manipolazioni avanzate sulla posta in entrata prima ancora di prelevarla dal server.

Il protocollo IMAP, che opera sulla porta 143, consente inoltre di:
- rinominare la propria casella elettronica;
- cancellare singoli messaggi senza essere costretto a prelevarli;
- leggere le intestazioni dei messaggi senza doverli prelevare interamente;
- prelevare addirittura solamente delle porzioni dei messaggi;
- effettuare l'acceso simultaneo alla stessa casella di posta.

La versione più recente, l'IMAP4, è candidata a diventare lo standard di riferimento su Internet e permette inoltre all'utente di archiviare i messaggi in cartelle direttamente sul server, di accedere contemporaneamente a più mail server e di condividere le mailbox con altri mail server.
Offre inoltre una migliore integrazione con la tecnologia MIME che viene usata per effettuare gli attachment ai messaggi.

Riassumiamo in uno schema i principali vantaggi e svantaggi dei due protocolli in modo da poterli confrontare.

POP3 (Post Office Protocol)

Vantaggi:
- alla connessione col server, il client scarica nel computer locale i messaggi liberando lo spazio occupato sul server di posta;

- la presenza dei messaggi off-line ne agevola la lettura senza necessità di rimanere connessi alla rete;
- quasi tutti i sistemi possono essere configurati per mantenere una copia di messaggi anche sul server dopo averli scaricati.

Svantaggi:
- alla connessione col server, il client scarica tutti i nuovi messaggi indipendentemente dalle loro dimensioni e dalla velocità di connessione;
- se l'utente accede al server da computer diversi, scarica la posta in parte su ciascun computer, con la scomodità di dover ricercare i documenti in tutti i suoi client;
- con una connessione lenta POP3 non offre le stesse prestazioni di IMAP.

IMAP (Internet Message Access Protocol)

Vantaggi:
- i messaggi rimangono sul server;
- l'accesso alla propria casella di posta può essere effettuato da postazioni diverse;
- permette la consultazione della sola intestazione prima di effettuarne lo scaricamento completo;
- è possibile effettuare il download dei messaggi per la consultazione in locale;
- permette di definire gerarchie di caselle postali per una migliore classificazione della posta;
- permette di associare ai messaggi uno o più marcatori per una successiva elaborazione;
- permette di effettuare ricerche per chiave sui vari messaggi o su un loro sottoinsieme.

Svantaggi:
- in caso di alto utilizzo della posta con allegati pesanti è possibile che si raggiunga la quota massima di utilizzo dello spazio sul server.

Ricordiamo che l'IMAP4, come già il POP3, si occupa solo della gestione della posta arrivata, e non della sua spedizione o della trasmissione in rete.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 In Internet, l'indirizzo di posta elettronica è del tipo:

a) nomeutente@dominio
b) dominio @destinatario
c) dominio @utente
d) indirizzo@dominio

2 MAU è l'acronimo di:

a) Mailbox User Acces
b) Mailbox User Agent
c) Mail Utent Agent
d) Mail Utent Acces
e) Mail User Agent

3 Quali tra le seguenti affermazioni sono false?

a) Il POP mail utilizza il protocollo SMTP per l'inoltro dei messaggi
b) Il POP mail utilizza il protocollo POP3 per l'inoltro dei messaggi
c) La web mail utilizza il protocollo SMTP per l'inoltro dei messaggi
d) La web mail utilizza il protocollo POP3 per l'inoltro dei messaggi
e) Il POP mail utilizza il protocollo SMTP per la ricezione dei messaggi
f) Il POP mail utilizza il protocollo POP3 per la ricezione dei messaggi
g) La web mail utilizza il protocollo SMTP per la ricezione dei messaggi
h) La web mail utilizza il protocollo POP3 per la ricezione dei messaggi

4 MIME è l'acronimo di:

a) Multimedia Internet Mail Extension
b) Multipurpose Internet Mail Extension
c) Mail Internet Multipurpose Extension
e) Multipurpose Internet Multimedia Extension
f) Mail Internet Mail Extension

5 Nella testata di un documento MIME si trovano i seguenti campi (indica quello errato):

a) MIME version
b) Content-Transfer-Encoding
c) Content-Type
d) Content-Definition
e) Content-ID
f) Content-Description

6 Ordina le fasi di una conversazione POP3.

a) .... TRANSACTION
b) .... AUTHORIZATION
c) .... UPDATE

Test vero/falso

1 L'indirizzo di posta elettronica è associato a una persona.
2 POP mail è sinonimo di Web mail.
3 La POP mail permette di leggere e inviare i messaggi di posta elettronica dal computer.
4 Outlook Express, Eudora, IncrediMail, Mozilla sono server mail.
5 Il protocollo SMTP è stato definito in RFC 812.
6 L'RCF 822 specifica il formato per i messaggi di posta come ASCII a 7 bit.
7 L'RFC 1822 descrive lo standard MIME per la trasmissione di dati multimediali.
8 base64 è un valore possibile per il Content-Type di MIME.
9 In base64 ogni carattere viene rappresentato con sei bit.
10 La conversazione POP3 utilizza la porta 100.
11 IMAP4 è l'evoluzione di POP3.
12 IMAP4 utilizza la stessa porta di POP3.



LEZIONE 6

DNS: IL DOMAIN NAME SYSTEM

IN QUESTA LEZIONE IMPAREREMO...

- le funzioni del DNS
- i compiti dei root server e dei server di dominio
- il formato dei messaggi DNS e dei record di risorsa

Generalità: nome simbolico e indirizzo IP

Per identificare un host oppure un router in Internet sappiamo che è necessario conoscere il suo indirizzo IP, che è unico sulla rete pubblica: ma un sito Internet è generalmente individuato da un nome mnemonico, che è più semplice utilizzare da parte dagli "esseri umani".

> È decisamente più facile ricordare www.hoepli.it piuttosto che il suo indirizzo IP 193.202.13.14.

Anche ai PC di una LAN viene generalmente associato un nome simbolico che però deve essere mantenuto in corrispondenza del rispettivo indirizzo IP: in una rete locale questo può essere fatto con un semplice "file hosts" composto da coppie di dati, come nel seguente esempio:

- 192.168.1.1 Contabilità_1
- 192.168.1.2 Contabilità_2
- 192.168.1.3 Magazzino_1
- 192.168.1.4 Magazzino_2
- 192.168.1.5 Vendite_1
- 192.168.1.6 Server
- …

ma al crescere della dimensione della rete e quindi del numero di host questa soluzione è impraticabile.

> Capita anche di assegnare allo stesso host più di un nome simbolico: siamo in presenza quindi di alias.

In Internet la risoluzione dei *nomi degli host* in *indirizzi IP* è un servizio che viene effettuato dal Domain Name System (DNS).



DNS

Con Domain Name System si intende:
- un *data base distribuito* che memorizza coppie (nome simbolico – indirizzo IP) su un insieme di nodi della rete (name servers);
- un *protocollo* a livello applicazione che regola la comunicazione tra hosts e name servers utilizzato da altri protocolli per *la risoluzione* dei nomi simbolici.

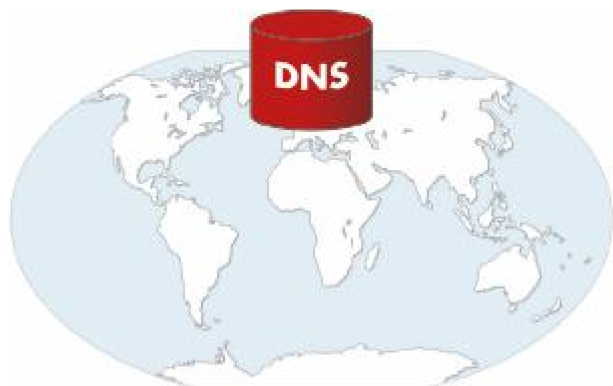
ESEMPIO 14

Se inseriamo in un browser (*HTTP client*) il nome:
http://www.hoepli.it/index.html
Il client HTTP attiva un client DNS che
- estrae dall'URL l'indirizzo simbolico IS del server (hoepli.it);
- interroga il DNS e ottiene l'indirizzo IP corrispondente a IS;
- invia quindi una richiesta (secondo il protocollo HTTP) al web server (passa l'indirizzo IP al livello TCP).

Funzioni e caratteristiche del DNS

Il servizio Domain Name System è definito in RFC 1034 e RFC 1035 e funziona attraverso lo scambio di messaggi UDP sulla porta 53.

Possiamo individuare tre funzioni principali del DNS:
1. traduzione dei *nomi simbolici* in indirizzi IP;
2. gestione degli alias, cioè permettere la definizione di *più nomi simbolici* per lo stesso host: per esempio può essere comodo utilizzare un soprannome al posto di un nome completo come www.rai.it al posto di rcsn1.roma.rai.it.
   Analogo discorso degli alias può essere fatto anche per gli indirizzi di posta elettronica: è più semplice utilizzare rossimario@rai.it piuttosto che rossimario@malsrv1.roma.rai.it
3. esecuzione del bilanciamento del carico: molti web server, come per esempio www.ebay.com, devono gestire una enorme quantità di richieste e per soddisfarle senza collassare le connessioni è necessario *replicare* i server in modo tale da avere più PC che contemporaneamente (in parallelo) possano offrire lo stesso servizio; allo stesso nome simbolico viene quindi associato un insieme di indirizzi IP ed è compito del DNS restituire gli indirizzi IP associati allo stesso nome simbolico secondo una disciplina circolare senza che gli utenti se ne accorgano ma distribuendo in egual misura le richieste su host differenti.



La prima idea per realizzare il DNS è quella di creare un enorme database e posizionarlo nel mezzo della rete Internet per fare in modo che tutti gli host lo "conoscano" e si rivolgano a esso.

È una soluzione che potrebbe anche essere realizzata, ma che comporterebbe un insieme di inconvenienti di seguito descritti che di fatto la rendono irrealizzabile:

- single point of failure: essendo un unico elemento diventerebbe un ◀ **SPOF** ▶ e un suo guasto sarebbe la causa di un "disastro universale";
- volume di traffico: tutto il traffico di Internet dovrebbe essere veicolato verso di esso, portando la rete al collasso;
- database distante: sarebbe irraggiungibile dagli host distanti (massimo numero di hop);
- manutenzione permanente: la manutenzione con l'aggiunta dei nuovi indirizzi e le modifiche di quelli esistenti renderebbero inagibile il sistema per la maggior parte del tempo.

Inoltre un database centralizzato su un singolo server DNS non è *scalabile*!



◀ **Single point of failure (SPOF)** A single point of failure (SPOF) is a potential risk posed by a flaw in the design, implementation or configuration of a circuit or system in which one fault or malfunction causes an entire system to stop operating. ▶

La soluzione adottata è stata quella di realizzare un database distribuito sul territorio utilizzando diversi server, ciascuno con la responsabilità di raccogliere, gestire, aggiornare e divulgare le informazioni che lo riguardano.
L'approccio è di tipo gerarchico dove gli elementi più alti nella gerarchia contengono molte informazioni non dettagliate e man mano si scende ai livelli inferiori aumenta il dettaglio delle stesse.
In prima approssimazione la gerarchia di server DNS è organizzata in tre classi:

- server radice;
- server top-level domain (TLD);
- server di competenza.

Quando un server locale non riesce a tradurre il nome simbolico di un sito Web in un indirizzo IP contatta un server DNS radice che se non conosce la mappatura contatta a sua volta un server DNS autorizzato ottenendo la mappatura e restituendola al server DNS locale.

Attualmente esistono sparsi per il mondo *13 server radice* etichettati da A a M organizzati come un cluster di server replicati, e hanno il nome da «a.root-servers.net» a «m.root-servers.net»; sono anche detti «**server di nomi radice**» e corrispondono ai domini di più alto livello (top-level domain).



I server TLD (Top-Level Domain) sono collegati agli altri server presenti nella rete generando una struttura "ad albero", e comprendono i domini com, org, net, edu ecc. e tutti i domini locali di alto livello, quali uk, fr, ca e jp.



NOMI DI PRIMO LIVELLO

I nomi di primo livello possono essere classificati in tre gruppi, a seconda della gerarchia di cui sono "a capo":

- gerarchia organizzativa:
  - .arpa corrisponde ai terminali derivanti dalla rete originale;
  - .com corrispondevano inizialmente alle aziende a vocazione commerciale.
  - .edu corrisponde agli enti educativi;
  - .gov corrisponde agli enti governativi;
  - .net corrispondeva inizialmente agli enti con tratti di rete;
  - ...
- gerarchia geografica:
  - .it domini italiani;
  - .fr domini francesi;
  - .uk domini inglesi;
  - .us domini degli Stati Uniti;
  - ...
- gerarchia generica:
  - .museum corrisponde ai musei;
  - .aero corrisponde all'industria aeronautica;
  - .biz (business) corrisponde alle aziende commerciali;
  - .coop corrispondente alle cooperative;
  - .info corrisponde agli enti con tratti di informazione;
  - .name corrisponde a nomi di persone o a nomi di personaggi immaginari.

Una seconda metodologia di classificazione "più formale" raggruppa i domini in due categorie:

1. i domini detti «generici», chiamati gTLD (*generic TLD*), classificati in base al settore di attività, a loro volta suddivisi in tre gruppi:
   - gTLD storici:
     - .arpa, .com, .edu, .gov, .int, .mil, .net, .org
     - gTLD nuovi introdotti nel novembre 2000 dall'ICANN:
     - .aero, .biz, .museum, .name, .info, .coop, .pro
   - gTLD speciali:
     - .arpa
2. I domini detti "nazionali", chiamati ccTLD (country code TLD) sono uno per ogni Paese e i loro nomi corrispondono alle abbreviazioni dei nomi degli Stati definiti dalla norma ISO 3166.



Zoom su...

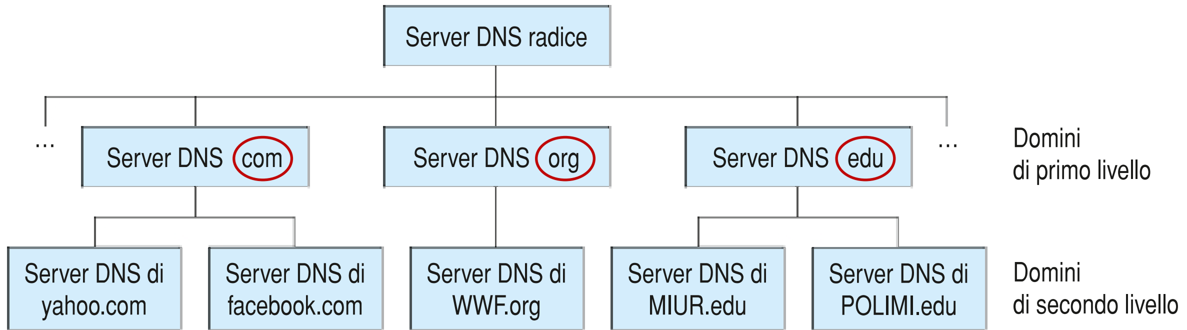
ICANN

ICANN è l'acronimo per Internet Corporation for Assigned Names and Numbers, una organizzazione senza scopo di lucro che aiuta a controllare la crescita di Internet: tra i suoi compiti è previsto anche quello di gestire il sistema dei nomi di dominio di primo livello generico (gTLD) e dei codici nazionali (ccTLD). La missione principale di ICANN è quella di assicurarsi che ogni dominio abbia un identificatore univoco (unico URL) e un unico indirizzo IP.

Al livello inferiore ogni nodo dell'albero viene detto "nome del dominio" e può essere al massimo lungo 63 caratteri: vengono definiti nomi di secondo livello e possono essere di due tipi:
- in alcune gerarchie prevedono semplicemente il nome dell'organizzazione, come *.enel.it*, *.rai.it* ecc.;
- in altre prevedono prima una caratterizzazione .bt.co.uk.

ESEMPIO 15

Un esempio di albero è il seguente:



L'estremità di un ramo (foglia) è detta host e corrisponde a un terminale o a un'entità di rete: il nome simbolico che viene attributo all'host deve essere unico nel dominio considerato (generalmente gli viene assegnato il nome www).

Oltre ai domini di secondo livello possono esistere altri livelli inferiori, chiamati sottodomini.

Il numero di livelli di sottodomini è variabile è può arrivare fino a un valore massimo di 127.

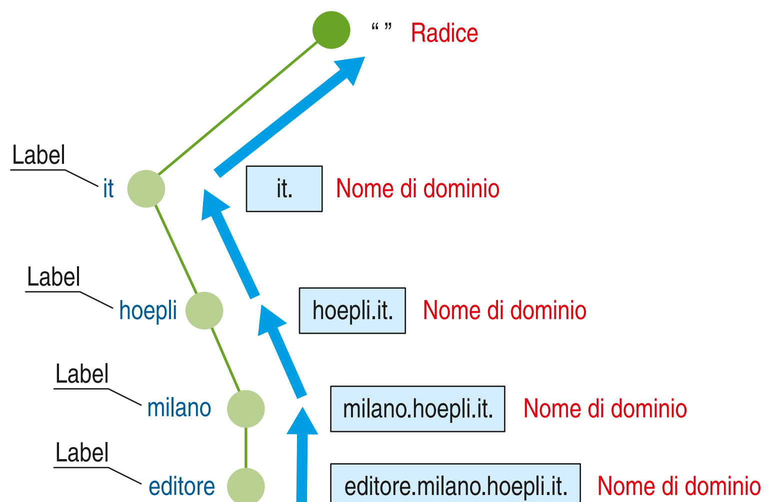
Anche ogni nodo dell'albero ha un nome di dominio che si ottiene dalla sequenza di etichette separate da punti (.) ottenute scorrendo il ramo dalla foglia alla radice, come per esempio www.milano.hoepli.it.; ogni punto separa un nodo e concatena da sinistra verso destra le label associate ai vari nodi a partire da quello in questione fino alla radice.

L'ultima etichetta è quella della radice, che è una **stringa nulla**: quindi tutti i **nomi di dominio terminano con un punto**.

ESEMPIO 16

Per ottenere il nome del dominio della foglia editore risaliamo il ramo fino alla radice inserendo dopo ogni etichetta che incontriamo il separatore ".".

Il nome di dominio della foglia è quindi editore.milano.hoepli.it.: possiamo anche avere nomi di dominio intermedi, a seconda del livello da cui partiamo, come hoepli.it. se consideriamo solo il secondo livello.



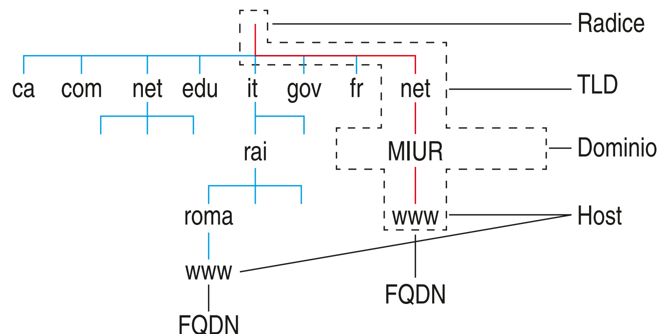


INDIRIZZO FQDN

Viene chiamato indirizzo FQDN, cioè Fully Qualified Domain Name o *Nome del Dominio Totalmente Qualificato*, il nome assoluto corrispondente all'insieme delle etichette dei

La lunghezza massima di un nome FQDN è di 255 caratteri e deve permettere di individuare inequivocabilmente un terminale sulla "rete di reti".

ESEMPIO 17



Il primo indirizzo FQDN indicato in figura www.MIUR.net. è così composto:



Il secondo indirizzo FQDN indicato in figura è www.roma.rai.it. e presenta un sottodominio:





INDIRIZZO PQDN

Se invece abbiamo una sequenza di stringhe che non terminano con una stringa nulla siamo in presenza di un Partial Qualified Domain Name o Nome del Dominio Parzialmente Qualificato.

La seguente tabella riporta alcuni indirizzi di dominio dei due tipi:

FQDN	PQDN
www.hoepli.it.	www.hoepli.it
istruzione.miur.it.	istruzione.miur
editore.hoepli.it.	editore.hoepli
www.mario.it.	www

I server di secondo livello prendono il nome di server di competenza o di dominio (domain server DNS o semplicemente domain server oppure server DNS) e ogni organizzazione che ha un host accessibile pubblicamente su Internet deve fornire i record DNS di pubblico dominio, che mappano i nomi di tali host in indirizzi IP.
Un server DNS si dice autoritativo per una determinata zona (◀ **Authoritative Server** ▶) se contiene tutti i dati relativi alle zone di sua competenza.



◀ **Authoritative DNS** Refers to an authoritative DNS server that receives domain/zone information (that is, the DNS resource records it is responsible for) by requesting this information from the primary server for the domain in a process known as a zone transfer. The main reason secondary servers were invented was to lower administrative burden. In other words, the administrator only has to maintain the primary server, after which the zone transfer mechanism ensures that the secondary servers are automatically updated. ▶

I record DNS, anche detti Resource Record (RR), sono memorizzati nei domain servers e, come vedremo in seguito, contengono un insieme di informazioni su quel domain name tra le quali la principale è l'indirizzo IP del dominio.

All'interno di un dominio, un server DNS può essere:
- primario: è il responsabile dell'inserimento e della eliminazione di informazioni relative ai nodi della rete tenendo aggiornate nel suo database le informazioni sulla corrispondenza tra nomi e indirizzi IP dei nodi appartenenti al proprio dominio di autorità;
- secondario: è una copia del database primario e quindi possiede le stesse informazioni del primario e viene aggiornato contemporaneamente a esso; viene interrogato dai client in caso di fallimento del primario.

Non è necessario che i server siamo fisicamente all'interno dell'organizzazione: possono essere mantenuti anche da service provider, che offrono un insieme di servizi quali la registrazione dei domini, il mantenimento (hosting) dei domini, oppure anche il semplice servizio di redirect o di gestione DNS.



Zoom su...

REDIRECT

Il servizio redirect permette di mostrare alla apertura di un dominio il contenuto di un sito web esistente su un altro dominio: è inoltre possibile scegliere se i visitatori possano o meno vedere l'indirizzo reale dello spazio Web sul quale risiedono le pagine.

Oltre questi server ogni ISP (università, azienda ecc.) ha un server DNS locale chiamato default name server e quando un host si connette a un ISP riceve un indirizzo IP e un DNS locale. Questo server non appartiene alla gerarchia ma gioca un ruolo fondamentale in quanto soddisfa tutte le richieste DNS degli host, opera cioè da proxy e inoltra la query nei livelli superiori della gerarchia di server DNS.

Ciascun host deve essere configurato con l'indirizzo del DNS server locale per il dominio a cui appartiene in quanto deve inviare verso di esso le proprie richieste di risoluzione: generalmente la configurazione avviene in maniera automatica al momento della sua connessione (ma può anche essere effettuata manualmente).

Risoluzione dei nomi di dominio

Il meccanismo che consiste nel trovare l'indirizzo IP corrispondente al nome di un host è detto "risoluzione di nome di dominio". Per effettuare questa operazione il client, che ha nella sua configurazione gli indirizzi IP dei due default name server del suo fornitore di accesso a Internet, il DNS primario (o principale) e il DNS secondario (o alternativo), invia una richiesta al server primario tramite il ◀ resolver ▶: il default name server consulta un database che contiene i nomi ed effettua la traduzione, inviando poi la risposta al client; se invece il server locale non riesce a trovare la corrispondenza nel suo database, consulta un server DNS di livello superiore.



◀ Resolver Si tratta di una applicazione generalmente integrata al sistema operativo che permette di realizzare la risoluzione di nome di dominio. ▶

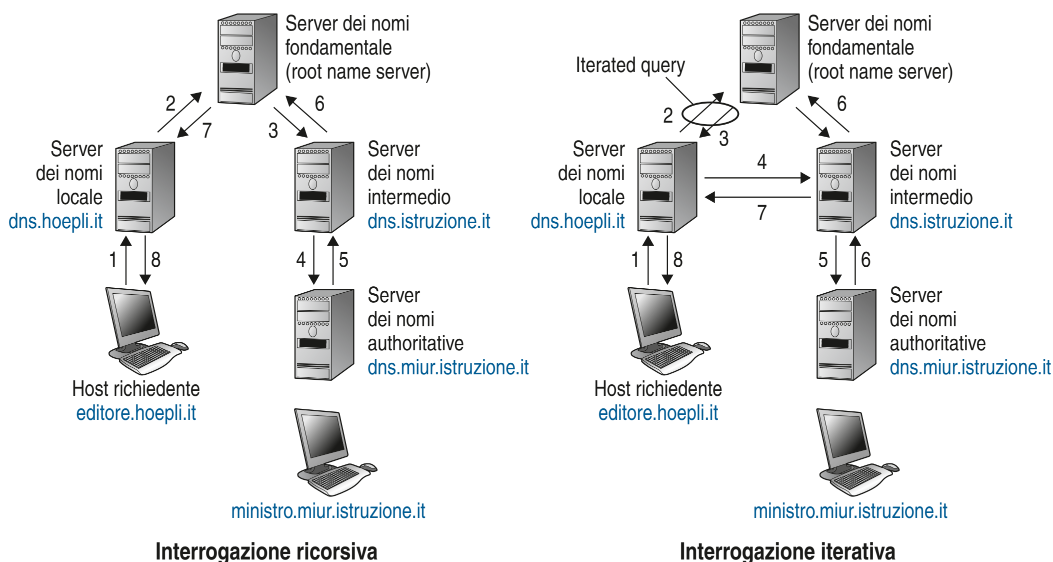
Ogni name server memorizza localmente i nomi usati di recente (mapping) insieme al riferimento del server remoto da cui l'informazione è stata ricavata: questo meccanismo prende il nome di caching, e viene realizzato per cercare di evitare che vengano spesso interpellati i server DNS di radice; in questo modo nei server DNS locali vengono memorizzati gli indirizzi dei server TLD e quindi la maggior parte delle volte si può risolvere il nome localmente, senza propagare la query.

Le informazioni della cache devono però periodicamente essere rinnovate (tipicamente entro due giorni): la procedura di aggiornamento è descritta nella RCF 2136.

Vediamo i passi per effettuare la risoluzione di un indirizzo richiesto da una applicazione che vuole connettersi a un host conosciuto con il suo nome di dominio (per esempio ministero.miur.istruzione.it):

1. la prima operazione consiste nell'interrogare il server di nomi primario definito nella sua configurazione di rete;
2. se l'informazione richiesta è presente nella sua cache oppure si verifica che il nome richiesto è relativo alla propria zona di autorità, la risposta viene marcata come non-authoritative e viene comunicata alla applicazione, altrimenti il server analizza la richiesta per poter interrogare il server radice che nel nostro caso è un server radice TLD «.it»:
   - se l'interrogazione è ricorsiva, contatta un server di livello superiore che possa risolvere il nome;
   - se è iterativa, si limita a comunicare al client il nome del server a cui rivolgersi.

Possiamo schematizzare graficamente le due situazioni:



Interrogazione ricorsiva   Interrogazione iterativa

Record DNS

Gli elementi che compongono il database dei DNS sono i record di risorsa (RR), che contengono le informazioni inerenti ai nomi dei domini.

La struttura del record di risorsa è la seguente:

Nome del dominio (FQDN)	TTL	Tipo	Classe	RData
hoepli.it.	3600	A	IN	163.5.255.85

Descriviamo i singoli campi.

- Nome del dominio: il nome del dominio deve essere un nome FQDN e quindi terminare con un punto, altrimenti il nome del dominio è relativo e il nome del dominio principale suffisserà il dominio inserito.
- Il campo TTL (*Time To Live*, tradotto in *speranza di vita*), indica la validità temporale del dato in quanto si è detto che la presenza del sistema di cache richiede di introdurre una scadenza alle informazioni in modo che i server intermedi possano regolarsi se accettare il dato o richiederne una verifica ai livelli superiori.
- Tipo: un valore a 16 bit che specifica il tipo di risorsa descritta dalla registrazione nel campo RDATA, tra le seguenti:
  - A: è il tipo di base che stabilisce la corrispondenza tra un nome canonico e un indirizzo IP;
  - CNAME (Canonical Name): permette di far corrispondere un alias al nome canonico;
  - HINFO: è un campo unicamente descrittivo che permette di descrivere soprattutto l'hardware (CPU) e il sistema operativo di un host;
  - MX (Mail eXchange): corrisponde al server di gestione della posta che viene interrogato dal server della posta in uscita quando un utente invia un messaggio elettronico a un indirizzo con formato: utente@dominio;
  - NS: corrisponde al server di nomi con autorità sul dominio;
  - PTR: un puntatore verso un'altra parte dello spazio di nomi dei domini;
  - SOA (Start Of Authority): il campo SOA permette di descrivere il server di nome con autorità sulla zona e l'indirizzo elettronico del contatto tecnico.
- Classe: la classe può essere
  - IN**:** corrispondente ai protocolli d'Internet;
  - CH: per i sistemi caotici e per applicazioni particolari.
- RDATA: sono dati attesi che cambiano a seconda del valore del campo tipo:
  - A: un indirizzo IP a 32 bits;
  - CNAME: un nome di dominio;
  - MX: un valore di priorità a 16 bits, seguito da un nome dell'host;
  - NS: un nome dell'host;
  - PTR: un nome di dominio;
  - SOA: più campi.

Possiamo riassumere tipo e RDATA in una tabella.

Tipo (sigla)	Significato	Contenuto RDATA
A	Indirizzo di host	Indirizzo IP a 32 bit
CNAME	Nome canonico	Nome di dominio canonico
HINFO	CPU & OS	Nome CPU e sistema operativo
MX	Scambiatore di posta elettronica	Nome del server di posta
NS	Server di nomi	Nome del server che ha autorità sul dominio
PTR	Puntatore	Nome di host
SOA	Inizio dell'autorità	Campi che specificano la gerarchia di nomi usata dal server

Messaggi DNS

I messaggi DNS sono le interrogazioni (query) che vengono effettuate dai client e le risposte che inviano i server: nel protocollo DNS le query e i messaggi di risposta hanno entrambi lo stesso formato.

Il tracciato è riportato nella seguente figura:

Identificazione	Flag	Header
Numero di domande	Numero di RR di risposta	
Numero di RR autorevoli	Numero di RR addizionali	
Domande		Sezione domande/risposte
Risposte		
Competenze		
Informazioni aggiuntive		

Possiamo individuare due sezioni:

1. l'header, composto da 12 byte, organizzato in sei campi:
   - identificazione: indice che numera con un valore binario a 16bit sia la domanda che la corrispondente risposta, in modo che possano essere associate;
   - flag: specificano le operazione richieste e le opzioni da applicare;
   - numero di domande/risposte RR: specifica il numero di richieste/risposta;
   - numero di RR autorevoli/risposte RR: specifica il numero di richieste autorità/informazioni di ritorno corrispondenti.
2. la sezione domanda o risposta, organizzata in quattro record (o gruppi di record):
   - I record: sezione di domanda di un DNS Query Message, che è così strutturata:

Nome richiesto	
Tipo interrogazione	Classe interrogazione

– Nome richiesto: è il nome del sistema su cui si richiedono informazioni;
– Tipo di interrogazione: è il tipo di informazione richiesta (indirizzo di un host, mail server);
– Classe interrogazione: è il protocollo associato alla richiesta.

All'atto della domanda il client riempie solo il primo record in quanto i successivi tre sono riservati a contenere le risposte da parte dei server:

- II record: sezione risposte (*answers*): contiene le informazioni restituite in risposta a una richiesta;
- III record: sezione competenze (*authority*): identifica il server che ha fornito l'informazione richiesta se il server di partenza ha dovuto contattarne un altro per soddisfare la richiesta (authority di competenza);
- IV record: sezione informazioni aggiuntive (*additional info*): contiene informazioni addizionali, che qualificano meglio una risposta.

Nel suo messaggio di risposta il server ricopia la sezione domanda e riempie le altre sezioni (risposta, competenze, informazioni aggiuntive) completandole con i dati letti nel resource record individuato. In un messaggio DNS possono essere presenti anche più domande contemporaneamente e, di conseguenza, nella riposta saranno replicati sia i record domanda che quelli risposta.

ESEMPIO 18

Vediamo un primo esempio di messaggio di richiesta (DNS Query Message):

Intestazione IP		
Identificatore = 4	Parametro = Richiesta + Ricorsione	Header
Numero richieste = 1	Numero risposte = 0	Header
Numero autorità = 0	Numero altre info = 0	Header
Nome: milano.hoepli.com.it Tipo = Host Classe = IN		Domanda

L'indirizzo di cui il client chiede la risoluzione è milano.hoepli.com.it e si indica al server che, se necessario, è possibile effettuare la ricorsione.

Vediamo come secondo esempio di messaggio il seguito che ha la richiesta appena fatta col messaggio precedente:

Intestazione IP		
Identificatore = 2	Parametro = Interrogazione	Header
Numero richieste = 1	Numero risposte = 0	Header
Numero autorità = 0	Numero altre info = 0	Header
Nome: milano.hoepli.com.it Tipo = Host Classe = IN		Domanda

Ipotizziamo che il server interrogato non abbia nessuna autorità sul nome e quindi a sua volta inoltri la query formulando la domanda a lui fatta come interrogazione a un altro server; si osservi che ha modificato l'identificatore assegnando un suo numero progressivo di riconoscimento per poi poter interpretare la risposta che gli perverrà.

Il successivo server interrogato è in grado di risolvere l'indirizzo e compila il seguente record:

Intestazione IP		
Identificatore = 2	Parametro = Risposta di autorità	Header
Numero richieste = 1	Numero risposte = 1	
Numero autorità = 0	Numero altre info = 0	
Nome: milano.hoepli.com.it Tipo = A (Host) Classe = IN		Risposta
Nome = puntatore al campo Nome della sezione domanda Tipo = A (Host) Classe = IN TTL = 20864 Lunghezza = 4 Risposta = 69.16.0.121		

In questo caso mantiene lo stesso valore dell'identificatore e comunica le informazioni relative al dominio, che verranno poi comunicate al client che le aveva richieste in un messaggio con identificatore uguale a 4, che è quello corrispondente alla query originale effettuata del client.

Intestazione IP		
Identificatore = 4	Parametro = Risposta di autorità	Header
Numero richieste = 1	Numero risposte = 1	
Numero autorità = 0	Numero altre info = 0	
Nome: milano.hoepli.com.it Tipo = A (Host) Classe = IN		Risposta
Nome = puntatore al campo Nome della sezione domanda Tipo = A (Host) Classe = IN TTL = 20864 Lunghezza = 4 Risposta = 69.16.0.121		

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Con Domain Name System si intende:

a) un data base distribuito che memorizza coppie di dati (nome simbolico – indirizzo IP)
b) un data base distribuito che memorizza coppie di dati (nome simbolico – indirizzo router)
c) un protocollo a livello applicazione che regola la comunicazione tra hosts e router
d) un protocollo a livello applicazione che regola la comunicazione tra router e name servers
e) un protocollo a livello applicazione che regola la comunicazione tra hosts e name servers

2 Le tre funzioni principali del DNS sono (indicare quella inesatta):

a) tradurre dei nomi in indirizzi
b) gestire gli alias
c) individuare i domini
d) eseguire il bilanciamento del carico

3 La soluzione di avere un unico database per il DNS è irrealizzabile perché:

a) diventa un single point of failure
b) il traffico lo farebbe collassare
c) avrebbe dimensioni troppo elevate
d) sarebbe spesso troppo distante e irraggiungibile
e) sarebbe troppo lento
f) sarebbe permanentemente in manutenzione

4 Quale tra i seguenti non è un dominio di alto livello?

a) info
b) rcf
c) usa
d) edu
e) biz
f) gov
g) it
h) rip
i) arpa
j) name

5 I domini generici sono classificati in tre gruppi (indicare quello errato):

a) storici
b) vecchi
c) nuovi
d) speciali

6 Con indirizzo FQDN si intende:

a) Finally Qualified Domain Name
b) Fully Query Domain Name
c) Fully Qualified Domain Name
d) Finally Query Domain Name

Test vero/falso

1 A ogni host corrisponde un unico indirizzo IP. V F
2 Un host locale non può avere più di un nome simbolico. V F
3 Il servizio DNS funziona attraverso lo scambio di messaggi TCP sulla porta 53. V F
4 Il DNS viene realizzato con un database gerarchico distribuito. V F
5 I server di nomi radice sono 15 etichettati da A a M. V F
6 La gerarchia organizzativa è una classificazione dei top level domain. V F
7 La sigla ccTLD significa city code top-level domain. V F
8 La lunghezza del nome di un dominio può al massimo essere di 127 caratteri. V F
9 Il numero di livelli di sottodomini è variabile e può arrivare fino a un valore massimo di 127. V F
10 L'etichetta della radice è la stringa ".". V F
11 La lunghezza del nome di un dominio può al massimo essere di 127 caratteri. V F
12 La lunghezza massima di un nome FQDN è di 255 caratteri. V F
13 Se una interrogazione è ricorsiva, il server DNS contatta un server di livello superiore. V F

Domande aperte

1 Descrivi la procedura di risoluzione di un nome in caso di interrogazione ricorsiva.

....

....

....

2 Descrivi la procedura di risoluzione di un nome in caso di interrogazione iterativa.

....

....

....

3 Indica la struttura di un record RR e fanne un esempio.

4 Completa la seguente tabella che riporta tipo e RDATA dei messaggi DNS.

Tipo (sigla)	Significato	Contenuto RDATA
A		
CNAME		
HINFO		
MX		
NS		
PTR		
SOA		

5 Completa il tracciato di un messaggio DNS.

Esercitazioni di laboratorio 1

I PROXY SERVER, LA NAVIGAZIONE ANONIMA E I COOKIES

Ricordiamoci che ogniqualvolta effettuiamo una connessione a Internet, il provider assegna un indirizzo IP al quale sono associate tutte le nostre azioni compiute nella rete. In linea di principio, soltanto i provider e le autorità giudiziarie possono associare un indirizzo IP a una persona fisica ma nella pratica lo può fare facilmente qualsiasi amministratore di un sito tramite, per esempio, i famigerati cookie. Per difendere il nostro anonimato e la nostra privacy su Internet basta innanzitutto camuffare l'indirizzo IP assegnatoci dal provider mostrandone uno diverso ai siti Web che visitiamo, come vedremo in questa esercitazione di laboratorio.

Per realizzare la ◀ navigazione anonima ▶ possiamo in estrema sintesi adottare tre metodi: uno è quello che fa uso di una particolare rete chiamata rete TOR, tramite la quale viene periodicamente modificato il nostro indirizzo IP, un altro utilizza particolari siti che reindirizzano la richiesta di una pagina web nascondendo il nostro indirizzo IP, il terzo infine utilizza un proxy.



◀ Navigazione anonima Indica una navigazione che tutela la privacy del client, garantendo che il server esterno, non possa conoscere l'indirizzo IP del client. Questa tecnica ha il principale scopo di impedire la connessione a quei siti che utilizzano l'indirizzo IP del client per scopi di autenticazione o di riconoscimento delle sessioni. ▶

La redirezione anonima

Esistono numerosi siti che offrono il servizio di proxy server online e che consentono infatti di schermare l'indirizzo IP dell'utente che ha effettuato la richiesta della pagina. Dall'esempio di figura possiamo notare che è sufficiente scrivere l'indirizzo del sito da raggiungere nella casella di testo denominata Web Address per raggiungere la pagina voluta in totale anonimato:



Nella figura notiamo come sia possibile raggiungere per esempio una pagina web non visibile a noi italiani: si tratta del sito di google.ca ▶.



Come possiamo notare facendo click sul pulsante Surf Anonymously il sito reindirizza la richiesta alla pagina indicata nascondendo l'indirizzo IP dal quale è partita la richiesta.

L'uso di un proxy

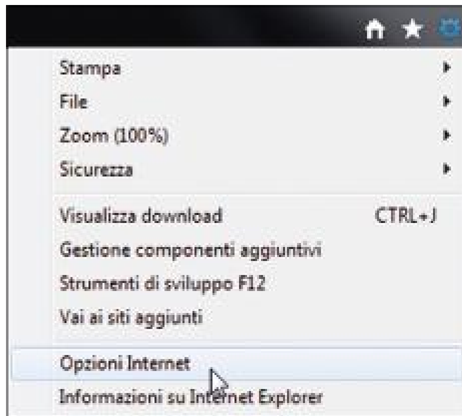
L'uso di un proxy è uno dei metodi che garantisce l'anonimato durante la navigazione. La procedura che segue mostra come impostare il browser per realizzare una navigazione anonima:

1. Per prima cosa dobbiamo impostare Internet Explorer per navigare sempre attraverso un proxy. Innanzi tutto abbiamo bisogno di un proxy server e del suo indirizzo. Al seguente indirizzo:
   http://spys.ru/en/anonymous-proxy-list/
   vi è un elenco dei proxy anonimi gratuiti disponibili sulla rete, la seguente figura mostra il sito indicato:

È buona norma cambiare spesso il proxy scegliendone uno nuovo dalla lista onde essere sicuri della sua efficacia.

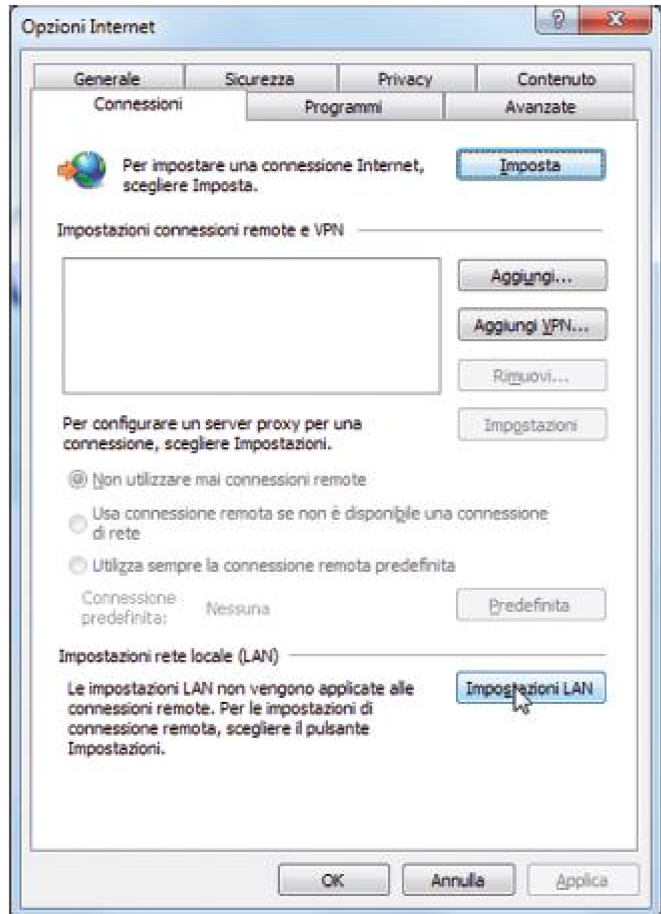
2. Scegliamo un proxy, in questo caso quello di indirizzo 216.168.96.25 con porta 8080

3 Apriamo il browser, in questo caso Internet Explorer, e selezioniamo l'icona delle impostazioni rappresentata da un ingranaggio ▼:



4 Selezioniamo la voce Opzioni Internet.

5 Adesso attiviamo la scheda Connessioni, quindi facciamo click sul pulsante Impostazioni LAN ▶:



6 Spuntiamo la casella Utilizza un server proxy, quindi inseriamo l'indirizzo IP del server proxy e la porta come indicato dalla figura ▶:



7 Dopo aver confermato su OK verifichiamo il funzionamento testando l'accesso a un sito.

La rete TOR

Per superare le limitazioni dei proxy, date soprattutto dalla possibilità di essere tracciati mediante i cookies, un gruppo di volontari sparsi per il mondo ha dato vita al progetto TOR (The Onion Routing). Hanno messo a punto una particolare rete di comunicazione che permette ai suoi utilizzatori di difendersi dalle ◀ analisi del traffico ▶.



◀ Analisi del traffico Si tratta di analisi che consistono nel monitorare le connessioni Internet per scoprire i siti visitati e avere notizie sulle ricerche effettuate su Internet così da poter fare delle analisi di mercato oppure semplicemente tenere sotto controllo il traffico sulla rete. ▶

L'idea che sta alla base del progetto è semplicissima: utilizzare più proxy insieme. In questo modo, le nostre richieste per il caricamento di un sito Internet passano attraverso almeno tre "intermediari" (detti relay) in modo tale che nessun malintenzionato in grado di spiare un solo punto del percorso possa capire dove sia diretta la richiesta e, soprattutto, da chi è stata effettuata.

I dati scambiati fra i diversi relay, inoltre, sono cifrati e ognuno conosce solo quale relay gli ha passato le informazioni e a quale inoltrarle, senza sapere chi ha dato vita alla richiesta di queste informazioni, cioè il nostro computer. In questo modo, se anche un relay venisse compromesso, sarebbe impossibile, a differenza dell'utilizzo di un singolo proxy, risalire alla nostra identità.

Tutta la rete TOR, inoltre, è strutturata in modo da modificare automaticamente ogni dieci minuti il percorso seguito dai nostri dati, per evitare che qualcuno possa collegare le azioni che noi compiamo su Internet.
La procedura seguente mostra come configurare TOR per la navigazione anonima.

1. Scarica l'ultima versione di TOR dal sito www.torproject.org, facendo click sul pulsante che ne consente il download ▶.



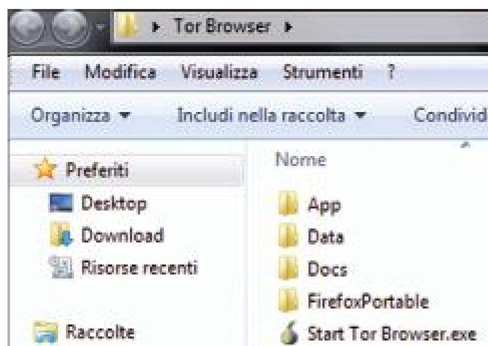
2. Seleziona la lingua e quindi fai click sul pulsante Download ▶.



❸ Dopo aver selezionato la directory destinazione, al termine del download ottieni il seguente file ▶.



❹ Dopo aver fatto doppio click sul file avviene l'estrazione dei file nella cartella denominata Tor Browser, che al termine conterrà i seguenti file ▶.

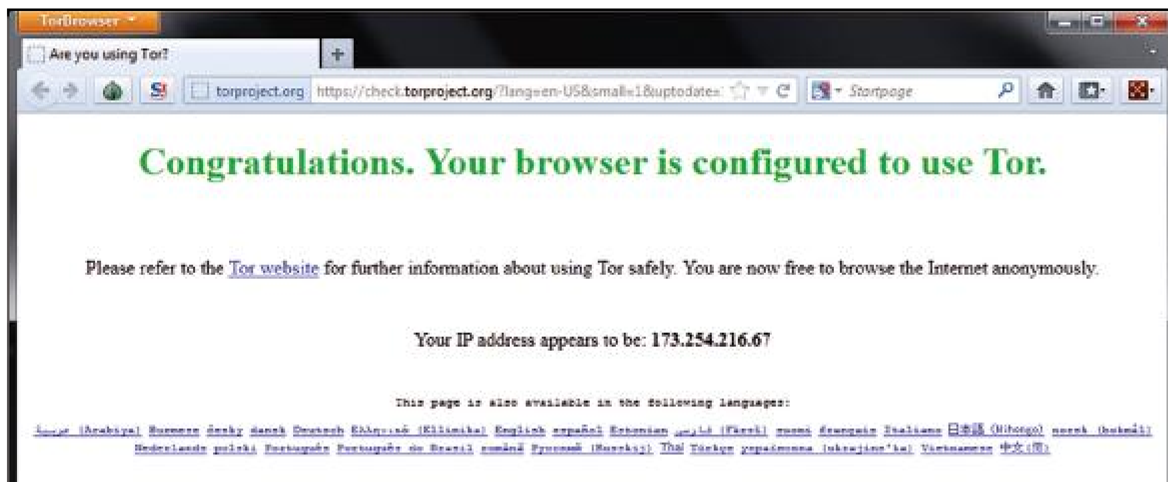


❺ Tra i file appare Start Tor Browser che consente di far partire il browser Firefox per la navigazione anonima. Fai doppio click sul file indicato, ottieni la seguente videata che mostra la connessione alla rete TOR ▶.



- Stato di TOR: permette di capire se TOR è in funzione sul nostro computer. Se tutto funziona correttamente, viene mostrata la dicitura *Tor è avviato* affiancata da un'icona di colore verde.
- Avvia/Ferma Tor: basta un clic per attivare o disattivare la protezione di TOR.
- Mostra la rete: in ogni momento possiamo vedere una mappa mondiale di tutti i computer collegati alla rete TOR e che, a loro volta, fungono da intermediari fra il nostro computer e i siti Internet.
- Avviamento del relaying: cliccando su questo pulsante possiamo entrare a far parte attivamente del progetto TOR, diventando a nostra volta intermediari per gli altri computer che usano la rete TOR.
- Usa una nuova identità: quando vogliamo, possiamo cambiare l'indirizzo IP con cui il nostro computer è identificato su Internet.
- Nascondi: premiamo questo pulsante per chiudere il pannello di controllo di TOR. Basta poi un clic sull'icona nella system tray per riattivarlo.

6 A questo punto si apre il browser che consente di navigare in modalità anonima:



Come possiamo notare nel browser viene indicato l'indirizzo IP che diverrà visibile al resto del mondo, in questo caso: 173.254.216.67.



Zoom su...

MIGLIORARE LA SICUREZZA CON FIREFOX

Se vogliamo davvero essere sicuri di non lasciare tracce su Internet, dobbiamo configurare correttamente le opzioni di sicurezza di Firefox. Accediamo al menu Strumenti, selezioniamo Opzioni, spostiamoci in Contenuti e togliamo il segno di spunta alla voce Attiva Javascript. Rimanendo nella finestra Opzioni di Firefox, spostiamoci nella scheda Privacy, togliamo il segno di spunta alla voce Accetta i cookie dai siti e clicchiamo su OK. Ora nella barra degli indirizzi del browser digitiamo about:config e premiamo Invio. Verrà caricato il registro di configurazione di Firefox. Digitiamo referer nel campo di testo Filtro e clicchiamo due volte sull'unica voce che viene mostrata. Nella finestra che si apre, digitiamo il numero 0 nel campo di testo e clicchiamo sul pulsante OK.

Esercitazioni di laboratorio 2

SMTP IN PRATICA: USO CON TELNET

In questa esercitazione invieremo un messaggio di posta elettronica utilizzando ◀ Telnet ▶. Per poter effettuare l'esercitazione dobbiamo innanzi tutto assicurarci di avere le porte 23 (Telnet) e 25 (SMTP) aperte in uscita. Inoltre molto spesso i firewall bloccano l'accesso a queste porte quindi potrebbe essere utile verificare che il firewall consenta l'accesso a queste due porte.



◀ Telnet è un protocollo di rete utilizzato su Internet. È solitamente utilizzato per fornire all'utente sessioni di login remoto di tipo linea di comando tra host su Internet. ▶

Gli amministratori di sistema utilizzano spesso Telnet per verificare se un server SMTP è attivo.

Iniziamo col dire che quando siamo connessi a un determinato provider non possiamo effettuare la connessione a un server SMTP di un altro provider, per esempio Alice con Tele2. I server SMTP non accettano utenti che non fanno parte della loro rete, per evitare problemi di spamming. Anche quando siamo connessi a un server SMTP del provider con il quale ci stiamo connettendo, non sempre è possibile l'invio della mail a utenti di altri provider.

Telnet è un servizio che Windows mette a disposizione, così come Linux. La procedura che segue illustra come attivare il servizio su Windows:

1. Prima di tutto dobbiamo aprire la finestra di prompt dei comandi, mediante cmd.exe dal menu esegui di Start:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>_
```

2. Il comando telnet attiva la finestra di dialogo di questo programma, come possiamo notare appare il prompt che indica Microsoft Telnet>:

```
C:\WINDOWS\system32\cmd.exe - telnet
Microsoft Telnet Client

Il carattere di Escape è 'CTRL++'

Microsoft Telnet>
```

3. Affinché venga effettuato l'eco dei comandi digitati sullo schermo dobbiamo eseguire il comando set localecho (set local echo in Windows 7):

```
C:\WINDOWS\system32\cmd.exe - telnet
Microsoft Telnet Client

Il carattere di Escape è 'CTRL++'

Microsoft Telnet> set localecho
Eco locale attivato
Microsoft Telnet>
```

4. Adesso proviamo a connetterci a un server SMTP; per fare questo dobbiamo ovviamente conoscere l'indirizzo del server. In questo esempio utilizzeremo il server di tin.it, si tratta di smtp.tin.it. Il comando che consente di attivare la connessione è open smtp.tin.it 25, dove 25 indica il numero del port dedicato alla mail. Il server risponde, se connesso, come indicato dalla figura seguente:

```
Telnet smtp.tin.it
220 vsmtp13.tin.it ESMTP Service ready
_
```

5. Dopo la connessione dobbiamo procedere con l'identificazione; per fare questo dobbiamo digitare il comando HELO seguito dal nome del dominio interessato. In questo caso il nome è rappresentato dal dominio del provider, quindi HELO tin.it:

```
Telnet smtp.tin.it
220 vsmtp8.tin.it ESMTP Service ready
HELO tin.it
250 vsmtp8.tin.it
_
```

L'elenco dei comandi TELNET è il seguente:

- VRFY <host name> verifica se un indirizzo è valido senza spedire
- EXPN <host name> espande i messaggi di una mailing list
- STAT controlla la dimensione e il numero dei messaggio
- TOP n° mess. n° righe. visualizza le prime righe del messaggio
- HELP <host name> visualizza i comandi accettati dal server
- 250 OK risposta affermativa dal server
- NOOP ritorno negativo dal server
- REPLY indica l'indirizzo email a cui rispondere
- CONNECT specifica il nome del provider a cui collegarsi e l'account da usare
- BINARY specifica un file di tipo binario
- GET specifica il nome del file da prelevare (con eventuali directory)

Esiste inoltre tutta una serie di comandi per la gestione dei messaggi secondo il protocollo SMTP:

Spedire email con Telnet	Ricevere le email con Telnet
telnet smtp.provider.it 25 invio helo provider.it (saluta il server) invio mail from:<indirizzo@mittente.it> invio rcpt to: indirizzo@destinatario.it invio data subject: oggetto + invio (lasciare una riga vuota) testo del messaggio + invio . (punto) invio quit (chiudi collegamento server) exit (chiudi dos)	telnet pop3.provider.it 110 invio user username pass password invio list (lista messaggi) invio retr + n° (visualizza messaggo) invio dele + n° (contrassegna messaggio) invio (così elimina il messaggio) rset (cancella) invio quit (chiudi collegamento server) exit (chiudi dos)

Al posto del comando HELO possiamo utilizzare EHLO come comando alternativo, il quale in genere mostra un elenco breve di comandi disponibili:

```
EHLO tin.it
250-vsmtp7.tin.it
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-AUTH=LOGIN
250-AUTH LOGIN CRAM-MD5 DIGEST-MD5 PLAIN
250-DELIVERBY 300
250 SIZE 31457280
```

6 Come abbiamo visto il server risponde con il messaggio: 250 vsmtp8.tin.it che indica l'avvenuta identificazione. Per autenticarci sul server è necessario includere dati utente e password espressi secondo il ◀ codice base64 ▶.



◀ Codice base64 È un sistema di numerazione posizionale che usa 64 simboli; è usato per codificare i dati riservati, come per esempio le autorizzazioni alla connessione (user name e password) normalmente espressi in formato Ascii. All'indirizzo http://ostermiller.org/calc/encode.html potete convertire il nome utente e la password in questo codice. ▶

A questo punto scriviamo la user name espressa in codice base64 seguita dall'invio:

```
Telnet smtp.tin.it
220 vsmtp8.tin.it ESMTP Service ready
EHLO tin.it
250-vsmtp8.tin.it
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-AUTH=LOGIN
250-AUTH LOGIN CRAM-MD5 DIGEST-MD5 PLAIN
250-DELIVERBY 300
250 SIZE 31457280
AUTH LOGIN
334 UXNlcm5hbWU6
bHVpZ2kubG9ydXNzb0B0aW4uaXQ=
334 UGFzc3dvcmQ6
_
```

7 Ora passiamo a inserire la password, sempre espressa in codice base64 seguita dall'invio:

```
Telnet smtp.tin.it
220 vsmtp8.tin.it ESMTP Service ready
EHLO tin.it
250-vsmtp8.tin.it
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-AUTH=LOGIN
250-AUTH LOGIN CRAM-MD5 DIGEST-MD5 PLAIN
250-DELIVERBY 300
250 SIZE 31457280
AUTH LOGIN
334 VXNlcm5hbWU6
bHVpZ2kubG9ydXNzb0B0aW4uaXQ=
334 UGFzc3dvcmQ6
cXdlcnR5dW1A
```

8 A questo punto il server risponde affermativamente se i dati corrispondono (LOGIN authentication successful):

```
Telnet smtp.tin.it
220 vsmtp13.tin.it ESMTP Service ready
EHLO tin.it
250-vsmtp13.tin.it
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-AUTH=LOGIN
250-AUTH LOGIN CRAM-MD5 DIGEST-MD5 PLAIN
250-DELIVERBY 300
250 SIZE 31457280
AUTH LOGIN
334 VXNlcm5hbWU6
bHVpZ2kubG9ydXNzb0B0aW4uaXQ=
334 UGFzc3dvcmQ6
cXdlcnR5dW1A
235 LOGIN authentication successful
```

9 A questo punto dobbiamo indicare l'indirizzo del mittente della mail che vogliamo inviare scrivendo MAIL FROM: seguito dall'indirizzo del mittente posto tra i segni di maggiore e minore (<>) e l'indirizzo del destinatario mediante il comando RCPT TO: seguito dall'indirizzo email del destinatario, sempre posto tra maggiore e minore (<>):

```
Telnet smtp.tin.it
220 vsmtp8.tin.it ESMTP Service ready
ehlo tin.it
250-vsmtp8.tin.it
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-AUTH=LOGIN
250-AUTH LOGIN CRAM-MD5 DIGEST-MD5 PLAIN
250-DELIVERBY 300
250 SIZE 31457280
AUTH LOGIN
334 VXNlcm5hbWU6
bHVpZ2kubG9ydXNzb0B0aW4uaXQ=
334 UGFzc3dvcmQ6
cXdlcnR5dW1A
235 LOGIN authentication successful
MAIL FROM:<elena.bianchi@tin.it>
250 MAIL FROM:<elena.bianchi@tin.it> OK
RCPT TO:<luigi.lorusso@tin.it>
250 RCPT TO:<luigi.lorusso@tin.it> OK
```

⑩ Attraverso il comando DATA indichiamo l'intenzione di scrivere il testo della mail. Dopo aver digitato DATA premiamo invio e quindi digitiamo il testo del messaggio:

```
Telnet smtp.tin.it
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-AUTH=LOGIN
250-AUTH LOGIN CRAM-MD5 DIGEST-MD5 PLAIN
250-DELIVERBY 300
250 SIZE 31457280
AUTH LOGIN
334 VXNlcm5hbWU6
bHVpZ2kubG9ydXNzb0B0aW4uaXQ=
334 UGFzc3dvcmQ6
cXdlcnR5dW1A
235 LOGIN authentication successful
MAIL FROM:<elena.bianchi@tin.it>
250 MAIL FROM:<elena.bianchi@tin.it> OK
RCPT TO:<luigi.lorusso@tin.it>
250 RCPT TO:<luigi.lorusso@tin.it> OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
Ciao, sono Elena come stai?
```

Per terminare la digitazione del testo del messaggio dobbiamo digitare, in sequenza: invio, punto (.) e ancora invio.



Zoom su...

ELENCO DEI SERVER SMTP PIÙ DIFFUSI DEI MAGGIORI PROVIDER ITALIANI

- LIBERO.IT mail.libero.it (anche per Libero ADSL)
- LIBERO.IT (dal 5-10-2009) smtp.libero.it (attivando richiesta di autenticazione)
- DOMINIOWEB.ORG mail.dominioweb.org
- INWIND.IT mail.inwind.it
- TELECOM.IT mail.tin.it / out.aliceposta.it o out.virgilio.it
- IOL.IT mail.iol.it
- INTERFREE.IT mail.interfree.it
- BLU.IT smtp.blu.it
- VODAFONE.IT smtp.net.vodafone.it / smtpmail.vodafone.it
- KATAWEB.IT smtp.katamail.com
- CIAOWEB.IT mail.ciaoweb.net /ciaosmtp.ciaoweb.it
- EMAIL.IT smtp.email.it
- TELE2.IT smtp.tele2.it
- SUPEREVA.IT mail.supereva.it
- INTERBUSINESS.IT mail.cs.interbusiness.it
- ALICE ADSL TELECOM out.aliceposta.it
- YAHOO.COM smtp.mail.yahoo.com
- YAHOO.IT smtp.mail.yahoo.it
- SUPEREVA.IT mail.supereva.it
- POSTE.IT relay.poste.it
- JUMPY.IT mail.jumpy.it
- FASTWEB.IT smtp.fastwebnet.it
- INFINITO.IT smtp.infinito
- EXCITE smtp.tiscali.it
- TELE2 EVERYDAY smtp.tele2.it
- TELE2 ADSL smtp.tele2.it
- LYCOS.IT smtp.lycos.it
- MONRIF.NET mail.monrif.net
- VODAFONE MAIL smtp.net.vodafone.it
- TIM.IT box.posta.tim.it
- ALICE.IT out.alice.it
- ALICE.BUSINESS mail.191.biz
- ROSSOALICE.IT out.aliceposta.it / mail.tin.it
- VIRGILIO.IT out.virgilio.it / smtp.virgilio.it
- MSN.COM pop3.email.msn.com
- VODAFONE.IT smtpmail.vodafone.it
- ALBACOM.IT relay.albacom.net / smtp.albacom.net
- EXCITE.IT smtp.tiscali.it
- LOMBARDIACOM.IT pop.lombardiacom.it
- TISCALI.IT smtp.tiscali.it / mail.tiscali.it

Esercitazioni di laboratorio 3

HTTP SNIFFING CON WIRESHARK

Scopo di questa esercitazione è quello di catturare due pacchetti e analizzarli individuando le caratteristiche del protocollo HTTP utilizzano il packet-sniffer Wireshark.

> Il pacchetto Wireshark è stato descritto nell'esercitazione di laboratorio Utilizzo di Wireshark del volume 1, che è anche reperibile tra le risorse on-line del presente volume sul sito www.hoepliscuola.it.

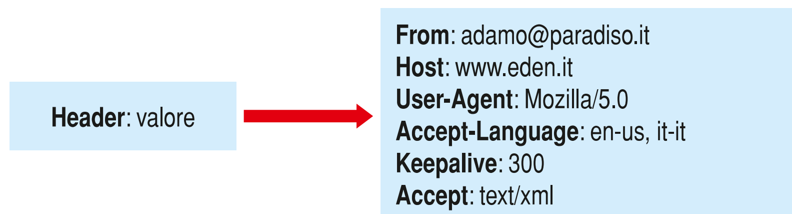
Generalità del protocollo HTTP

È il protocollo fondamentale del World Wide Web (WWW) descritto nella RFC 1945 (versione HTTP/1.0) nelle RFC 2068 e RFC 2616 (versione HTTP/1.1).
Ha la caratteristica di essere un protocollo Client-Server completamente testuale che utilizza la porta 80: è di tipo Stateless ma il server può implementare tecniche per mantenere lo stato quali i cookies, le sessioni, campi nascosti nei form, URL re direction ecc.

Il client invia una richiesta contenente un metodo al server, composta da tre parti:
- riga di richiesta (*request line*);
- intestazione (*header*);
- corpo del messaggio(*body*) (*opzionale*).

Nella riga di richiesta si trovano i metodi http, come GET, POST, HEAD, PUT, DELETE ecc.

Gli header forniscono informazioni sul contenuto della richiesta e hanno il seguente formato:



> Se un messaggio HTTP contiene un **body**, a seguito dell'header devono essere specificate le proprietà nell'header stesso.

A seguito della richiesta il server invia una risposta che ha una struttura analoga a quella della domanda: la riga di richiesta prende il nome di status line e i messaggi sono codificati con numeri di tre cifre dove la cifra delle centinaia li raggruppa come riportato nella tabella seguente:

Codice	Significato
1xx:	informazione
2xx:	successo
3xx:	redirezione (la richiesta è corretta ma è stata rediretta a un altro server)
4xx:	errore lato client (richiesta errata)
5xx:	errore lato server (c'è un problema interno del server)

Al codice segue sempre un messaggio di testo, generalmente in lingua inglese, che aiuta l'utente alla comprensione della risposta.



Zoom su...

STATUS CODE

Di seguito è riportato a titolo di esempio un elenco parziale di status code definiti nell'RFC 2616: l'elenco completo è reperibile all'indirizzo http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.

- 301 Moved Permanently: la risorsa è stata spostata in un'altra locazione in maniera permanente;
- 304 Not Modified: la risorsa non è stata modificata dall'ultima volta, non serve rinviarla;
- 400 Bad Request: il server non è in grado di capire la richiesta;
- 404 Not Found: la risorsa richiesta non è disponibile (per esempio, è stata richiesta una pagina inesistente);
- 500 Internal Server Error: c'è stato un errore interno nel server (per esempio, un errore di connessione al db).

Alla status line fa seguito un insieme di header e un body che contiene la risposta, che sono i contenuti della richiesta al server (pagina HTML, immagini, filmati ecc.).

L'interazione di base: richiesta/risposta

Come prima parte di esercitazione sul protocollo HTTP scarichiamo alcuni file da un dominio web, catturiamo con Wireshark i pacchetti che vengono scambiati tra il nostro host e il server e li analizziamo sempre con Wireshark.

> Se non avete la possibilità di catturare i pacchetti potete analizzare quelli disponibili tra le risorse on-line nella sezione dedicata a questo libro nel sito hoepliscuola.it, nella cartella risorse, memorizzati nei file HTTP-file1.pcap e HTTP-file2.pcap.

Iniziamo a esplorare il protocollo HTTP scaricando un semplice file HTML molto piccolo, che non contiene nessun altro oggetto o contenuto multimediale (immagine, suono ecc.).

❶ Manda in esecuzione Wireshark e setta come filtro http, come in figura ▶:



❷ Utilizzando il tuo browser connetteti a Internet alla pagina

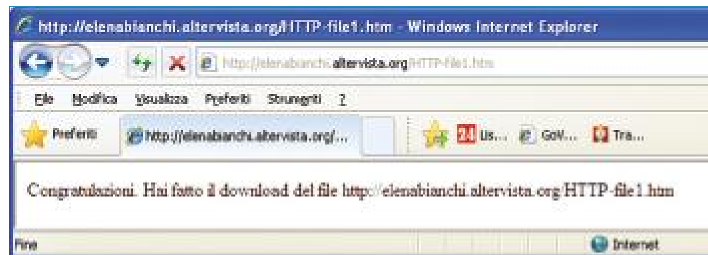
http://elenabianchi.altervista.org/HTTP-file1.htm

oppure alla pagina

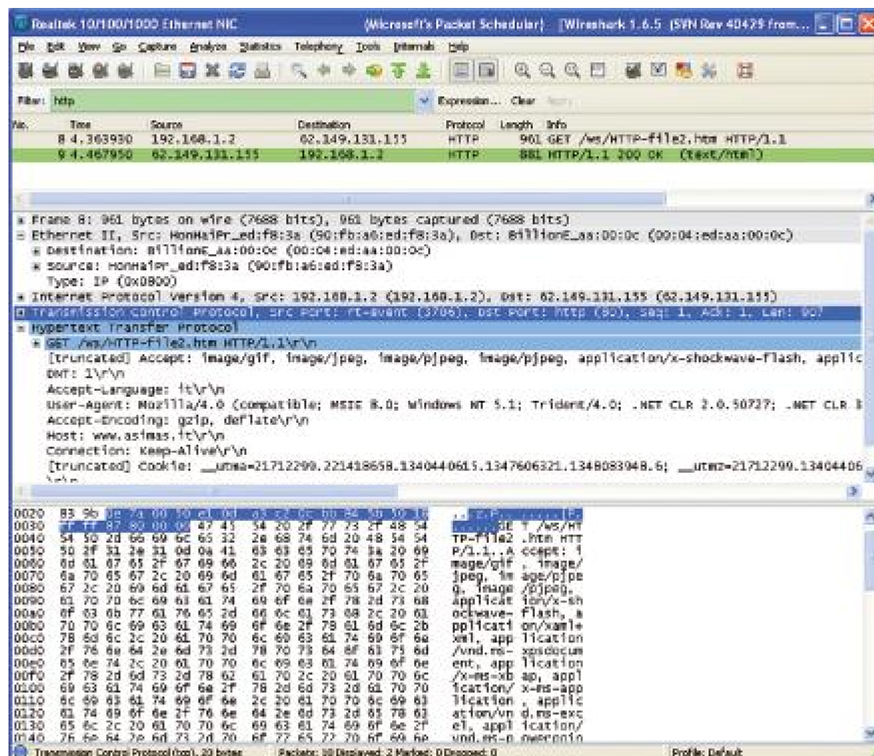
http://www.hoepliscuola.it//media/File/esercizi/5025-3/es01/HTTP-file1.htm

In questo modo nella finestra dell'elenco dei pacchetti vengono visualizzati solo i pacchetti contenenti messaggi http.

❸ Nel browser viene visualizzata una pagina HTML con una sola riga, come indicato nella figura ▶:



❹ Interrompi la cattura dei pacchetti: la finestra di Wireshark dovrebbe essere simile a quella riportata di seguito.



Sono presenti due messaggi HTTP: il messaggio GET, che ha inviato la richiesta dal vostro browser al server, e il messaggio di risposta verso il vostro browser.
La finestra centrale, quella di dettaglio, mostra informazioni del messaggio selezionato (nel caso di figura il primo): sono presenti le informazioni dell'intero frame Ethernet, dato che il messaggio HTTP è trasportato all'interno di un segmento TCP che a sua volta è all'interno di un datagramma IP, che è contenuto nel frame Ethernet.
Per espandere solo i dati che ci interessa analizzare basta cliccare sui box a fianco delle righe settando con [+] la riga desiderata e [-] quelle che si vogliono compattare.

Nella finestra potrebbero anche essere presenti messaggi relativi al file favicon.ico: con questo messaggio HTTP GET il vostro browser potrebbe chiedere automaticamente di visualizzare la piccola icona che viene visualizzata alla sinistra della riga di indirizzo della finestra del browser.





Prova adesso!

Rispondi alle seguenti domande, stampando i messaggi corrispondenti per meglio indicare le risposte:

1. Quale versione di HTTP sta usando il tuo browser?
2. Quale versione di HTTP è in esecuzione sul server?
3. Il tuo browser è in grado di accettare e, quindi, di interpretare qualche linguaggio?
4. Qual è l'indirizzo IP del tuo computer?
5. Qual è l'indirizzo IP del server?
6. Qual è il codice di stato restituito dal server al tuo browser?
7. Quando è stato modificato l'ultima volta il file che hai recuperato dal server?
8. Se individui che il file è stato modificato qualche minuto prima del tuo download, prova a riscaricarlo e confronta nuovamente l'ora dell'ultima modifica. Cosa puoi osservare? Perché?
9. Quanti byte sono stati inviati al tuo browser?

Il server e il browser continuano a scambiarsi messaggi: Wireshark li intercetta ed è possibile visualizzarli cliccando col tasto destro su uno qualunque dei pacchetti che fanno parte della connessione TCP e selezionare Follow TCP Stream.

Recuperare documenti di grandi dimensioni

Come secondo esercizio ci proponiamo ora di analizzare il protocollo http mentre scarichiamo un file di dimensioni abbastanza grandi da richiedere più invii.

La maggior parte dei browser mantiene una cache con gli ultimi oggetti scaricati: prima di eseguire il prossimo esercizio assicuratevi che la cache del vostro browser sia vuota
- per IE, selezionate Strumenti → Elimina cronologia esplorazione → Elimina.
- per Firefox, selezionate Strumenti → Cancella cronologia recente → Cancella adesso.

1. Avvia il browser web e svuota la cache.

2. Manda in esecuzione Wireshark.

3. Immetti la seguente URL nel browser:

   http://elenabianchi.altervista.org/HTTP-file2.htm

   oppure il seguente indirizzo

   http://www.hoepliscuola.it//media/File/esercizi/5025-3/es01/HTTP-file2.htm

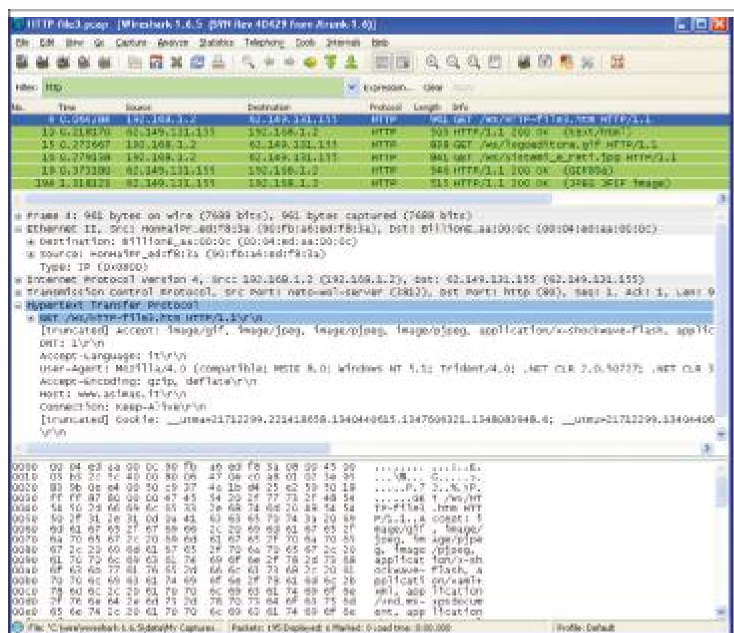
4. Il tuo browser visualizza la seguente pagina contenente due immagini ▶.



5. Interrompi la cattura pacchetti di Wireshark.

6. Filtra i messaggi inserendo come nell'esercizio precedente "http" nella finestra del filtro: si ottiene una situazione simile a quella riportata nella figura ▶.

   Nella finestra sono ora presenti più righe: seleziona la risposta alla prima GET (seconda riga) e osserva nella finestra dei dettagli quelli relativi al protocollo http individuando la riga denominata "Reassembled TCP Segments".



Il protocollo TCP ha spezzato il file richiesto in più parti, dato che la sua dimensione è di circa 4Kbyte e quindi è superiore al valore massimo per MTU di 1500 byte imposti da Ethernet: la risposta viene quindi frammentata in alcuni pacchetti tra loro separati dove il primo contiene la dicitura “Reassembled TCP Segments” per indicare che sarà seguito dai rimanenti pacchetti che formano l’intero messaggio HTTP.



Prova adesso!

Rispondi alle seguenti domande, stampando i messaggi corrispondenti per meglio indicare le risposte:

1. Quanti messaggi HTTP GET sono stati inviati dal vostro browser?
2. A quali indirizzi Internet sono state inviate queste richieste?
3. Quanti segmenti contenenti dati sono stati necessari per trasportare l’unica risposta HTTP?
4. Qual è la dimensione di ogni singolo segmento?
5. Qual è il codice di stato e la frase associata alla risposta del server?
6. Ci sono righe di stato HTTP trasmesse nei pacchetti TCP di risposta successivi al primo?
7. Come sono state scaricate le due immagini dal vostro browser?
8. Come si riesce a comprendere che appartengono alla pagina da voi richiesta?

Esercizi proposti

Esercizi proposti

1. Prova a ottenere dal web server un messaggio di risposta con la Status Line 400 Bad Request.
2. Prova a ottenere dal web server un messaggio di risposta con la Status Line 404 Not Found.
3. Scarica dal web un file immagine di dimensione superiore a 20 Kbyte e analizzane il pacchetti.
4. Collegati a un sito che richiede una autenticazione e cerca di individuare, analizzando i pacchetti scambiati tra client e server, i dati di autenticazione (user name e password).

Esercitazioni di laboratorio 4

DNS e Nslookup

Il Domain Name System (DNS), come descritto nella lezione 6, traduce nomi di host in indirizzi IP, e in questa esercitazione analizzeremo cosa avviene alla richiesta di un client di risoluzione di un indirizzo: sappiamo che i server DNS gerarchici comunicano tra di loro ricorsivamente o iterativamente per rispondere alla query, in modo del tutto trasparente al client che ha posto la domanda.

Il programma Nslookup

I sistemi operativi Linux/Unix e Microsoft mettono a disposizione il comando nslookup che ci permette di inviare una query DNS al server DNS specificato e di visualizzare la risposta ricevuta.
Basta posizionarsi sulla linea di comando e digitare nslookup per visualizzare la seguente schermata:

```
C:\WINDOWS\system32\nslookup.exe
Server predefinito:  fmidmzdns007.dsl.opitel.it
Address:  91.80.36.136

>
```

Per uscire e terminare l'interrogazione si digita il comando EXIT.

Come si può vedere nella seguente videata, è possibile interrogare sia un server radice, un server di dominio top-level, un server di fiducia (authoritative) oppure un server DNS intermedio.

```
C:\WINDOWS\system32\nslookup.exe
Server predefinito:  fmidmzdns007.dsl.opitel.it
Address:  91.80.36.136

> www.hoepli.it
Server:  fmidmzdns007.dsl.opitel.it
Address:  91.80.36.136

Risposta da un server non di fiducia:
Nome:    dh50.teligo.net
Address:  82.145.162.97
Aliases:  www.hoepli.it

> _
```

Il comando può anche essere eseguito con l'opzione type=NS: in questo caso viene richiesto di specificare i server di competenza (authoritative) per il dominio ricercato.

In questo esempio il client è localizzato a Milano e il server DNS locale è fmidmzdns007.dsl.opitel.it.



Prova adesso!

Spesso i firewall presenti nella rete delle aule di informatica non permettono di interrogare un server DNS a scelta ma solo quello locale, mentre dal "PC casalingo" è possibile effettuare tutte le interrogazioni.

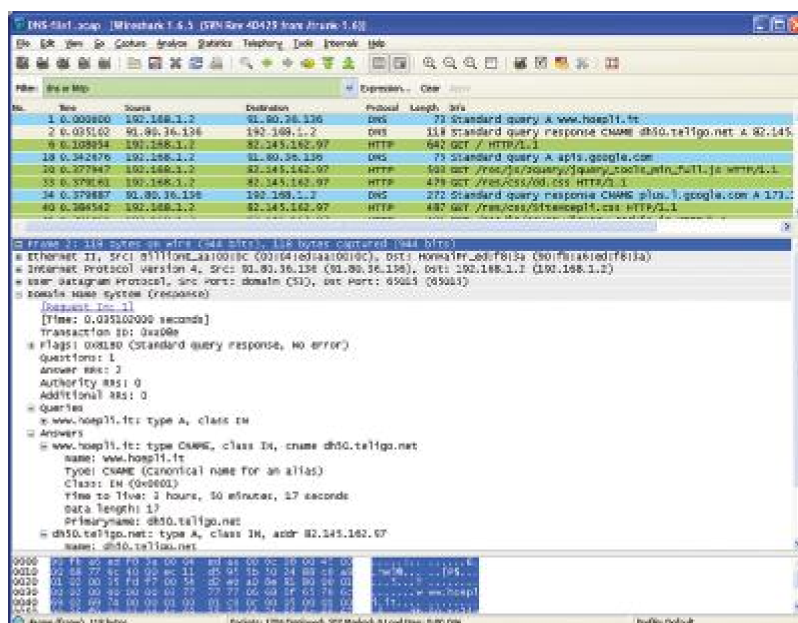
Rispondi alle seguenti domande, stampando i messaggi corrispondenti per meglio indicare le risposte:

1 Quanti messaggi HTTP GET sono stati inviati dal tuo browser?
2 A quali indirizzi Internet sono state inviate queste richieste?
3 Quanti segmenti contenenti dati sono stati necessari per trasportare l'unica risposta HTTP?
4 Qual è la dimensione di ogni singolo segmento?
5 Qual è il codice di stato e la frase associata alla risposta del server?
6 Ci sono righe di stato HTTP trasmesse nei pacchetti TCP di risposta successivi al primo?
7 Come sono state scaricate le due immagini dal tuo browser?
8 Come si riesce a comprendere che appartengono alla pagina da te richiesta?

Tracciare il protocollo DNS con Wireshark

Cattura ora i pacchetti che vengono scambiati con Wireshark.

1. Apri il browser e, per precauzione, svuota la cache.
2. Manda in esecuzione Wireshark e immetti "dns or http" nel filtro (in minuscolo) per visualizzare solo i pacchetti relativi ai protocolli DNS e HTTP.
3. Inizia a effettuare la cattura dei pacchetti con Wireshark.
4. Visita la pagina web http://www.hoepli.it.
5. Interrompi la cattura dei pacchetti.





Prova adesso!

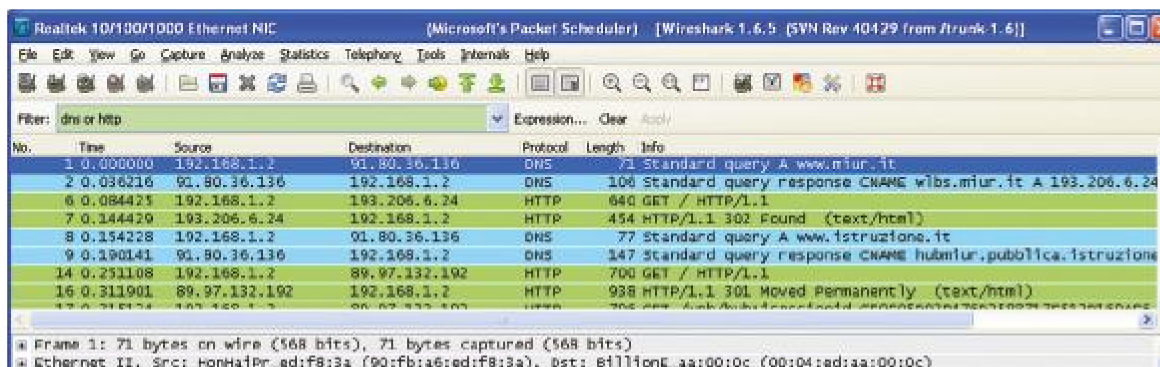
Rispondi quindi alle seguenti domande.

1. Le query DNS e i messaggi di risposta con che protocollo sono inviati?
2. Qual è la porta di destinazione per le query DNS?
3. Qual è la porta sorgente per i messaggi DNS di risposta?
4. Individua l'indirizzo IP del tuo server DNS locale mediante il comando ipconfig.
5. Individua a quale indirizzo IP è stata mandata la query DNS.
6. Confronta i due indirizzi e commenta il risultato.
7. Individua il "Type" della query DNS.
8. Individua la risposta DNS: cosa contengono i record di risposta?
9. Questa pagina web contiene immagini. Prima di recuperare queste immagini, sono state inviate altre query DNS?

Ripeti ora le operazioni per la cattura dei pacchetti:

1. Apri il tuo browser e, per precauzione, svuota la cache.
2. Manda in esecuzione Wireshark e immetti "dns or http" nel filtro (in minuscolo) per visualizzare solo i pacchetti relativi ai protocolli DNS e HTTP.
3. Inizia a effettuare la cattura dei pacchetti con Wireshark.
4. Visita la pagina web http://www.MIUR.it.
5. Interrompi la cattura dei pacchetti.

Dovresti ottenere una videata simile alla seguente:





Prova adesso!

Rispondi quindi alle seguenti domande.

1. Quante query sono state inviate?
2. Individua la porta di destinazione per il messaggio DNS di richiesta.
3. Individua la porta di destinazione per il messaggio DNS della risposta.
4. La richiesta DNS è stata mandata al tuo DNS locale? Scrivi l'indirizzo.
5. Individua il "Type" della query DNS.
6. Individua la risposta DNS: cosa contengono i record di risposta?

Esercitazioni di laboratorio 5

SMTP & POP: SNIFFING CON WIRESHARK

Generalità sul Simple Mail Transfer Protocol (SMTP)

Il Simple Mail Transfer Protocol (SMTP) è il protocollo standard per la trasmissione di email via Internet. È un protocollo client/server relativamente semplice, testuale, nel quale vengono specificati uno o più destinatari di un messaggio e, verificata la loro esistenza, il messaggio viene trasferito.
Il protocollo SMTP usa il protocollo di trasmissione TCP e, per accedervi, la porta 25: per l'invio della posta elettronica è necessario avere installato un software client come Opera 10, Mozilla Thunderbird, Outlook Express, The Email Client, Eudora ecc.
Sendmail fu uno dei primi (se non proprio il primo) programma a implementare il protocollo SMTP: fino a oggi sono stati scritti oltre 100 programmi che implementano il protocollo SMTP come client (mittente dei messaggi) o server (destinatario del messaggio).

Non basta avere istallato il software per spedire la posta in quanto è necessario poter accedere SMTP che inoltrerà i nostri messaggi al destinatario: i fornitori di connettività provvedono a fornire i parametri e le credenziali per poter effettuare l'autenticazione al server SMTP in modo da permettere ai propri clienti di inviare posta elettronica attraverso i programmi client, ma solo mediante un indirizzo connesso al proprio dominio.
Nella figura è riportata la scheda di configurazione del gestore di telefonia teletu.



La posta spedita viene memorizzata nel server che provvede a effettuare la consegna al server che gestisce la casella di posta (mailbox) del/dei destinatario/i.

L'SMTP è un protocollo che permette soltanto di inviare messaggi di posta, ma non di richiederli a un server: per fare questo il client di posta deve usare altri protocolli, quali il POP3 (Post Office Protocol) e l'IMAP (Internet Message Access Protocol).

Poiché SMTP è un protocollo testuale basato sulla codifica ASCII, non è permesso trasmettere direttamente testo composto con un diverso set di caratteri e tantomeno file binari: a tale scopo si utilizza lo standard MIME, che permette di estendere il formato dei messaggi mantenendo la compatibilità col software esistente.

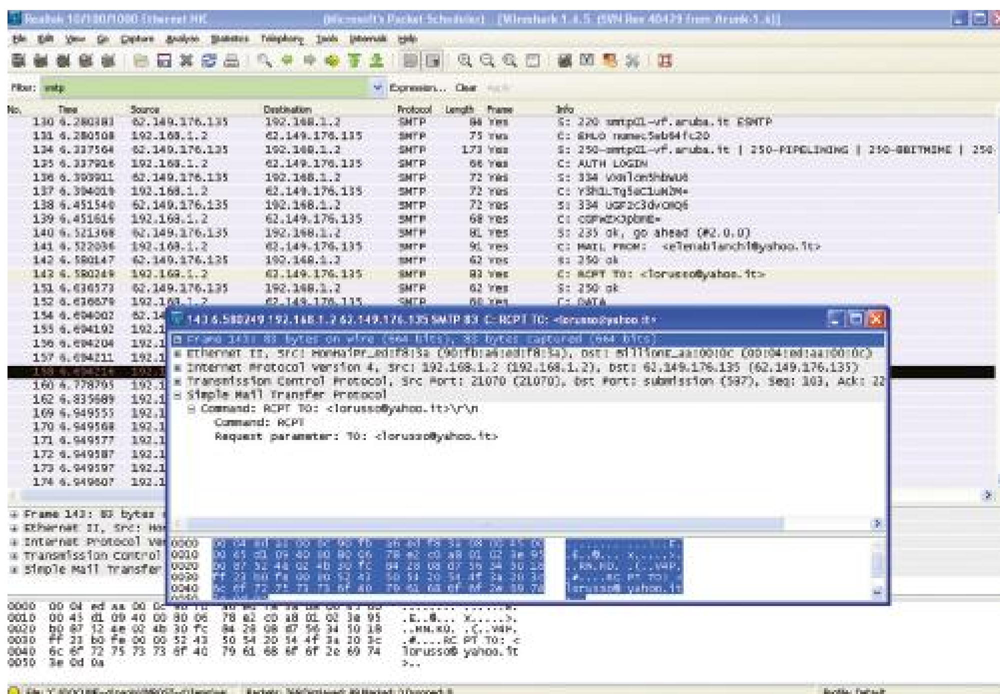
Cattura di pacchetti SMTP

Per poter verificare il corretto invio del messaggio email è necessario possedere un indirizzo email reale al quale poter accedere per visionare il messaggio di posta inviato.

Per poter capire il funzionamento del protocollo SMTP si effettuerà un invio di email utilizzando un client dedicato: spesso i laboratori scolastici non permettono di effettuare questa attività e quindi si consiglia di effettuare questa esercitazione sul PC domestico e di salvare i pacchetti per poi analizzarli successivamente in laboratorio.

1. Avvia un client di posta elettronica e prepara un messaggio: si consiglia di inviare un messaggio email a un proprio account esistente e attivo, dal quale si possa poi, tramite webmail, controllarne l'avvenuta ricezione;
2. avvia Wireshark e inizia a effettuare la cattura dei pacchetti;
3. invia il messaggio di posta;
4. termina la cattura e salva i pacchetti che sono stati intercettati in un file (per esempio, salva i pacchetti in smtp1.pcap);
5. filtra i pacchetti ricevuti inserendo smtp (in minuscolo) nella linea Filter.

A video si presenta una schermata come quella riportata di seguito:





Prova adesso!

1. Individua le righe dove avviene la autenticazione da parte del server: cosa si può osservare?
2. Qual è il significato del codice 220 inviato dal server al client?
3. Evidenzia il mittente e il destinatario del messaggio.
4. Qual è il server di posta in uscita? Dove è possibile individuarlo?
5. Quali sono le dimensioni del messaggio inviato?
6. Individua il pacchetto che contiene l'oggetto del messaggio.
7. Cosa si può osservare in merito alla privacy?
8. Ora ripeti le operazioni di invio componendo un nuovo messaggio che comprende un allegato (per esempio, una immagine con dimensioni maggiori di 10 Kbyte).
9. Come varia il traffico catturato con Wireshark?
10. Come viene gestito l'invio dell'allegato?

Il Post Office Protocol (POP)

Il Post Office Protocol è un protocollo che permette alle persone autenticate di accedere a un account di posta elettronica presente su di un host per scaricare le email del relativo account utilizzando un software di posta (come per esempio Outlook Express).
Come per l'invio dei parametri per la lettura della posta sono forniti dal server che offre il sevizio e vengono inseriti nella configurazione dell'account del destinatario della posta: la ricezione dei messaggi sul PC locale è generalmente permessa da tutti i gestori del servizio di comunicazione, anche per account registrate su altri domini.

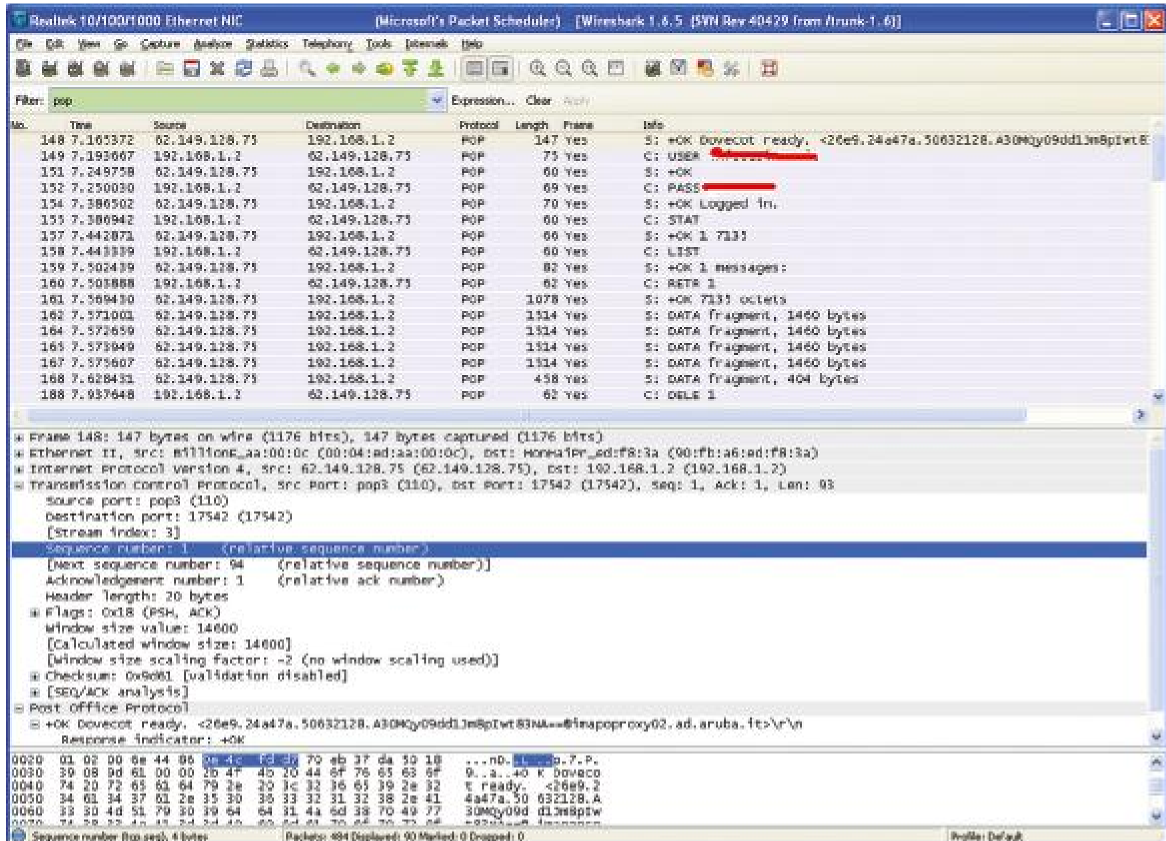
> Il protocollo POP3 non prevede alcun tipo di cifratura, quindi le password utilizzate per l'autenticazione fra server e client passano in chiaro: l'evoluzione del protocollo POP che utilizza MD5 è APOP.

1. Manda in esecuzione Wireshark in modo da catturare i pacchetti;
2. avvia il client di posta elettronica per esempio, Outlook express;
3. quando la posta è scaricata, ferma Wireshark: dovresti ricevere "almeno" i due messaggi prima spediti durante l'esercizio sul SMTP;



4. termina la cattura e salva i pacchetti che sono stati intercettati in un file (per esempio, salva i pacchetti in smtp2.pcap);
5. filtra i pacchetti ricevuti inserendo pop (in minuscolo) nella linea Filter.

A video dovresti avere una situazione simile a quella di figura:



> È anche possibili isolare i messaggi usando un filtro con l'indirizzo IP: nel nostro esempio inserendo
>
> ip.addr==192.168.1.2 // indirizzo del mittente
>
> oppure
>
> ip.addr==62.149.128.75 // indirizzo destinatario

Se dalla rete scolastica fosse impossibile effettuare l'esecuzione di un programma client di posta elettronica puoi scaricare il file smtp2.pcap dalla cartella materiali dello spazio web riservato a questo volume all'indirizzo www.hoepliscuola.it



Prova adesso!

1. Individua le righe dove avviene la autenticazione da parte del server: cosa si può osservare?
2. A che porta del server il client si è connesso per ricevere il messaggio?
3. Evidenzia il mittente e il destinatario del messaggio.
4. Dove è possibile individuare il server di posta?
5. Quali sono le dimensioni del messaggio inviato?
6. Individua il pacchetto che contiene l'oggetto del messaggio.
7. Ora ripeti le operazioni analizzando i pacchetti ricevuti per il secondo messaggio, quello contenente l'allegato.
8. Come varia il traffico catturato con Wireshark?
9. Come viene gestita la ricezione dell'allegato?
10. Che tipo di encoding è usato per l'allegato?
11. Quanti bit vengono usati per l'encoding?

5 IL SISTEMA OPERATIVO GNU/LINUX

UNITÀ DI APPRENDIMENTO

OBIETTIVI

- Riconoscere le caratteristiche del S.O. Linux
- Riconoscere il ruolo del boot loader e del kernel Linux
- Conoscere la gestione del file system nei sistemi GNU/Linux
- Saper distinguere le distribuzioni Linux in base alle caratteristiche principali
- Conoscere come avviene la redirezione dei comandi da terminale
- Saper riconoscere le partizioni e il montaggio delle stesse in un sistema Linux

ATTIVITÀ

- Installare il sistema operativo Linux per la distribuzione Ubuntu
- Applicare i comandi principali della shell Linux
- Saper montare e smontare un dispositivo
- Saper gestire il file system
- Saper utilizzare le directory e la redirezione dei comandi
- Gestire i file, le directory, gli utenti e i permessi in un sistema operativo Linux Ubuntu

LEZIONE 1

L'AVVIO DEL SISTEMA

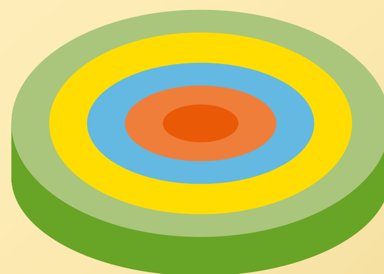
IN QUESTA LEZIONE IMPAREREMO...

- a conoscere come avviene l'avvio del sistema
- a definire il ruolo dei boot loader
- a individuare le partizioni da Linux
- a conoscere il boot loader Grub

L'avvio del sistema

Nei sistemi a PC il Sistema Operativo (S.O.) sovrintende a tutte le operazioni del sistema, e il suo ruolo coinvolge la sincronizzazione e la messa in esecuzione di altri moduli oltre alla gestione delle periferiche, come per esempio la gestione della concorrenza tra i processi che utilizzano la stessa CPU. I sistemi operativi che gestiscono attività concorrenti sono chiamati sistemi Operativi Multitasking.

I sistemi operativi devono poter essere installati in una partizione di un disco per poter essere utilizzati. Una partizione è uno spazio del disco fisso in cui possiamo installare un sistema operativo oppure salvare semplicemente dei files. Generalmente un nuovo disco non possiede partizioni, per cui ne deve essere creata almeno una.



Se in un disco abbiamo più partizioni, ciascuna di esse verrà trattata come se fosse un disco separato, in grado di memorizzare dati di tipo diverso che non interferiscono tra di loro come, per esempio due sistemi operativi distinti. Infatti due o più S.O. non possono coesistere nella stessa partizione.
I sistemi operativi tipo Windows, devono essere obbligatoriamente installati in una partizione primaria, mentre per Linux non c'è differenza. Una partizione estesa o logica differisce dalla partizione primaria perché a sua volta può essere ulteriormente partizionata.

Alcuni sistemi operativi possono essere avviati e utilizzati direttamente da una unità CD/DVD, mediante una connessione di rete ethernet con il supporto di un server (PXE - PrebooteXecution Environment), oppure da un supporto di memoria esterno, come per esempio una pen drive USB. In tal caso il programma che sovrintende l'avvio del sistema operativo, il ◀ BIOS ▶, dovrà consentire l'avvio da dispositivo ottico o da porta USB.



◀ BIOS Acronimo di Basic Input Output System, ed è un insieme di routine software, memorizzate su memoria ROM, oppure FLASH, che fornisce una serie di funzioni di base per l'accesso all'hardware e alle periferiche integrate nella scheda madre da parte del sistema operativo e dei programmi. Possiamo sintetizzare il suo esercizio nelle due operazioni fondamentali: POST (Power On Self Test) e Boot. ▶



Zoom su...

SEQUENZA DI BOOT

Il BIOS possiede una interfaccia utente che consente di scegliere la sequenza di boot, ossia di selezionare da quale unità effettuare l'avvio del S.O., in genere attraverso la pressione del tasto Canc durante l'accensione del PC. Una schermata tipica di avvio del BIOS è la seguente:



In questa finestra possiamo individuare la sezione denominata BootSequence che consente la modifica della sequenza delle unità periferiche dalle quali effettuare l'avvio del S.O. In questo caso possiamo notare che vi sono due lettori DVD, quindi un Hard Disk:



Il funzionamento è assai semplice, il sistema effettua l'avvio del sistema operativo secondo l'ordine impostato: il S.O. che verrà avviato sarà il primo utile che è stato trovato nell'unità indicata.

La maggior parte dei sistemi operativi non consente tuttavia l'avvio diretto da supporto di memoria, è pertanto necessario installarli sul computer. Ogni S.O. possiede una propria procedura di installazione, in genere mediante una interfaccia grafica. Dopo la fase di installazione un utente può interagire con il S.O. con l'interfaccia grafica in modalità GUI, in modalità linea di comando oppure mediante le chiamate ai servizi di sistema.
La fase di avvio di un sistema operativo viene anche chiamata fase di bootstrap ed è quel processo di caricamento dei file necessari al funzionamento del sistema da unità di memoria a disco a memoria RAM. In tal modo nella memoria RAM risiederà quella parte del S.O. pronta all'esecuzione. Il programma che sovrintende l'operazione di caricamento viene chiamato boot loader e, come abbiamo visto, è parte del BIOS. Possiamo schematizzare la sua funzione come segue:

- Il BIOS carica il boot loader;
- il boot loader carica l'immagine del kernel del S.O., oppure i boot sectors di altri sistemi operativi, come nel caso di Windows, passando ad altri programmi la successiva operazione di boot;
- il boot loader indica alla CPU di eseguire il kernel;
  – quando è caricato, il kernel, si trova nella fase di esecuzione e provvede all'avvio del processo chiamato init;
- il boot loader termina l'esecuzione e il controllo viene passato al S.O. che inizierà la fase vera e propria di caricamento.

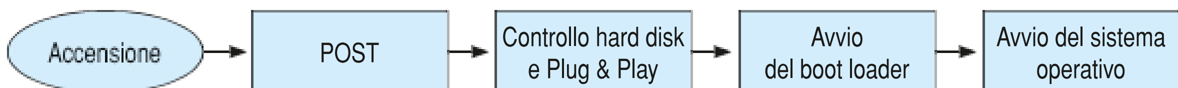
La procedura di avvio del sistema

Abbiamo visto che il BIOS dopo aver effettuato il POST effettua il ◀ bootstrap ▶.
La procedura di avvio del sistema può essere eseguita in seguito a un avvio a freddo (cold boot), quando il sistema viene avviato premendo il pulsante di accensione presente sul case, oppure in seguito a un riavvio a caldo (warm boot), quando si riavvia in modo software. Nel primo caso (avvio a freddo) viene effettuata la procedura completa, nel secondo caso senza il POST.



◀ Bootstrap È il processo che viene effettuato per passi successivi: prima si fanno dei semplici passi, poi, sfruttando questi, se ne fanno altri sempre più complessi. Il termine bootstrap deriva dal detto anglosassone "Pulling yourself up by the bootstraps", e significa "Alzarsi a partire dai lacci delle scarpe". ▶

Le fasi principali che caratterizzano l'avvio di un PC possono essere così schematizzate:



Vediamo le singole fasi.

1. All'accensione del PC il ◀ firmware ▶ del sistema verifica sommariamente la funzionalità degli altri circuiti principali, situati sulla stessa motherboard.
A quel punto la CPU esegue una routine di inizializzazione del sistema, la cui prima istruzione si trova all'indirizzo di memoria convenzionale FFFF0h. La cella di memo-



◀ Firmware Si tratta di software memorizzati su circuiti integrati a sola lettura, a differenza del software che viene generalmente memorizzato in file, posti su memoria di massa. ▶



Circuito del firmware.

ria posta a tale indirizzo effettua una istruzione in linguaggio macchina di salto incondizionato (Jump) a una routine chiamata appunto BIOS, memorizzata in un apposito circuito integrato di sola lettura (ROM, EPROM o FLASH) sulla motherboard.

2 Le routine del BIOS effettuano quindi il POST (Power On Self Test), ovvero la verifica del funzionamento di base dell’hardware del sistema. Se durante tale operazione viene trovato un malfunzionamento, la procedura si arresta ed emette un suono diverso a seconda del codice di errore ◀ **post error** ▶ riscontrato.



◀ **Post Errors** When you switch on or reset you PC it performs a diagnostic test called a Power-On Self Test (POST) to check that all the components are present and working correctly. First, it checks core components such as the system clock, processor, RAM, keyboard controller and graphics card. If any device fails this part of the POST, you’ll hear series of beeps sound from the desktop computers. After the graphics card has passed its test, the BIOS can then indicate any errors onscreen, such as the classic “Keyboard error or no Keyboard present. Press F1 to continue” message. When the POST has completed successfully, most PC emit a short beep to let you know the hardware is working correctly. That’s why beep codes differ depending on the bios manufacturer computer, so refer to your bios mainboard manual if you have it. Below are the list of common beep codes for AMI BIOS: ▶

1 beep	Everything is OK
2 beeps	Memory parity error (reseat or replace memory)
3 beeps	Memory read or write error (reseat or replace memory)
4 beeps	Motherboard timer is not working properly
5 beeps	Processor or memory error (reseat or replace processor and memory)
6 beeps	Keyboard controller failure (replace motherboard)
7 beeps	Processor exception interrupt error (reseat or replace processor)
8 beeps	Display memory read or write failure (reseat or replace graphic card)
9 beeps	BIOS ROM Checksum errors (replace BIOS chip motherboard)
10 beeps	Bad cache memory (replace cache memory if possible)
1 long, 3 short beeps	Memory error (reseat or remove any memory recently added and reseat all other memory)
1 long, 8 short beeps	Graphic card error (reseat graphics card)

In seguito viene eseguita una routine di inizializzazione della scheda video, in genere contenuta in memoria a partire dall’indirizzo assoluto C000h, che mostra informazioni sullo schermo. Quindi provvede a ricercare ed eseguire i driver specifici di altri dispositivi, come per esempio il driver del controller ATA, situato sempre in memoria convenzionale a partire dall’indirizzo C8000h. Viene quindi visualizzata la schermata di avvio relativa al BIOS e verificata la funzionalità della memoria RAM, indicando a video la quantità di memoria esaminata.



3 Le routine del BIOS controllano il circuito di gestione degli interrupt (PIC - Programmable Interrupt Controller), riconoscono gli hard disk e i dispositivi Plug & Play (se il BIOS supporta questa funzionalità) e visualizzano quindi un prospetto della configurazione del sistema.

4 Viene avviato il boot loader che effettua le seguenti operazioni:

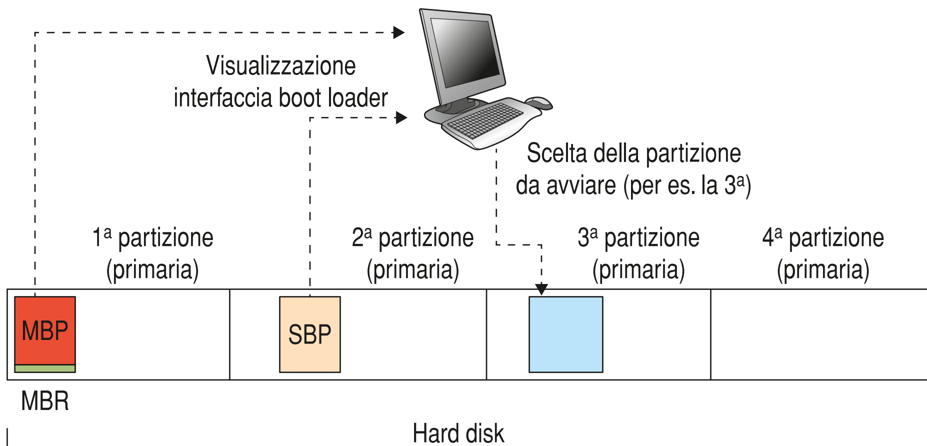
- in base alla sequenza di boot viene verificato ciascun dispositivo indicato per assicurarsi della sua presenza e del fatto che sia effettivamente avviabile, caratteristica riconosciuta nei primi byte del suo ◀ MBR ▶;
- viene caricato in memoria, a partire dall'indirizzo 7C00h, il primo settore del primo disco avviabile, che rappresenta l'MBR del disco e viene eseguito il codice in esso contenuto, chiamato MBP (Master Boot Program);
- l'MBP carica la seconda sezione del boot loader, chiamata SBP (Slave Boot Program), che prepara l'interfaccia utente per la scelta del sistema operativo da avviare;
- dopo che l'utente ha scelto il sistema operativo, l'SBP avvia il processo di caricamento del sistema: nel caso di Linux procede al caricamento e all'avvio del kernel (il nucleo del sistema operativo) che a sua volta lancerà in esecuzione il processo iniziale necessario per il funzionamento del sistema.



◀ MBR Significa Master Boot Record, contiene le informazioni sulle partizioni dell'unità a disco e i relativi indirizzi fisici espressi in traccia e settore, di inizio e fine. Inoltre contiene per ciascuna partizione un particolare flag che indica se è avviabile. ▶

Il boot loader

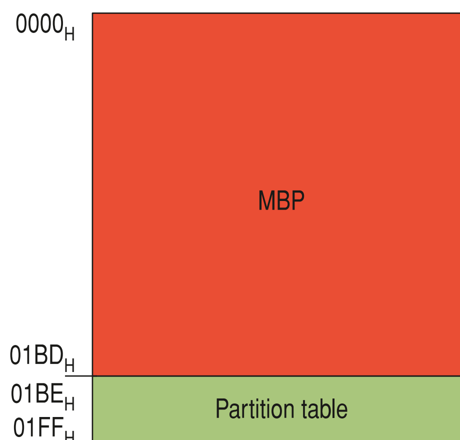
Il boot loader si compone sostanzialmente, come abbiamo visto sopra, dell'MBP (Master Boot Program) e del SBP (Slave Boot Program). A causa della ridottissima disponibilità di memoria l'MBP esegue in sostanza solo il caricamento dell'SBP, il quale permette di scegliere il sistema operativo, o la partizione da avviare e procede quindi al suo avvio.



L'MBP (Master Boot Program) è un codice eseguibile che si trova nei primi 446 byte del Master Boot Record. Viene avviato dal BIOS con un interrupt 19H, subito dopo le operazioni di POST. Si occupa di scansionare le partizioni, trovare e avviare il rispettivo boot loader che deve essere eseguito. L'MBT (Master Boot Table) è una tabella che si trova subito dopo l'MBP. La sua dimensione è 64 byte. Contiene informazioni sulle partizioni (tipologia, bootable, CHS iniziale, CHS finale, numero e dimensione cluster). Quando abbiamo delle partizioni estese, viene allocato su ciascuna di esse un nuovo record chiamato EBR (Extended Boot Record) all'interno del primo settore di ogni partizione logica. La struttura è identica all'MBR, ma lo spazio riservato al Master Boot Program risulta inutilizzato. L'EBT contiene il settore di inizio della partizione logica e il puntatore alla partizione logica successiva.

Il Master Boot Record

Il master boot record è il primo settore di un disco, posizionato sul cilindro 0, testina 0, e settore 1. Contiene, come abbiamo visto in precedenza le informazioni essenziali per l'avvio del sistema, lo schema logico della sua composizione è indicato a fianco: ▶



L'MBR ha una dimensione pari a quella minima di una traccia, cioè di 512 byte, di cui i primi 446 byte sono destinati a contenere il MBP (Master Boot Program) e i 66 byte rimanenti per la Partition Table. La Partition Table contiene le informazioni relative al partizionamento del disco, in particolare quattro elementi chiamati entry, di 16 byte l'uno, che descrivono le partizioni primarie. Questo è il motivo per cui possiamo avere al massimo quattro partizioni primarie, o 3 primarie e una estesa. Ciascun entry contiene le informazioni riportate nella tabella seguente:

Posizione del byte	Significato
0	Indicatore di boot, contiene il valore 80h se la partizione è avviabile
1-3	Testina, settore e cilindro del primo blocco della partizione
4	Identificatore del file system contenuto nella partizione
5-7	Testina, settore e cilindro dell'ultimo blocco della partizione
8-11	Numero di blocchi prima della partizione
12-15	Numero di blocchi che costituiscono la partizione

È importante notare che la dimensione di un disco è data da:

(numero di cilindri) × (numero di testine) × (63 settori/traccia) × (512 byte/settore)

Per esempio, un ipotetico hard disk con 3148 cilindri, 16 testine, 63 tracce, possiede la seguente dimensione:

(3148 cilindri) × (16 testine) × (63 tracce) × (512 byte/settore) = 1,624,670,208 Byte

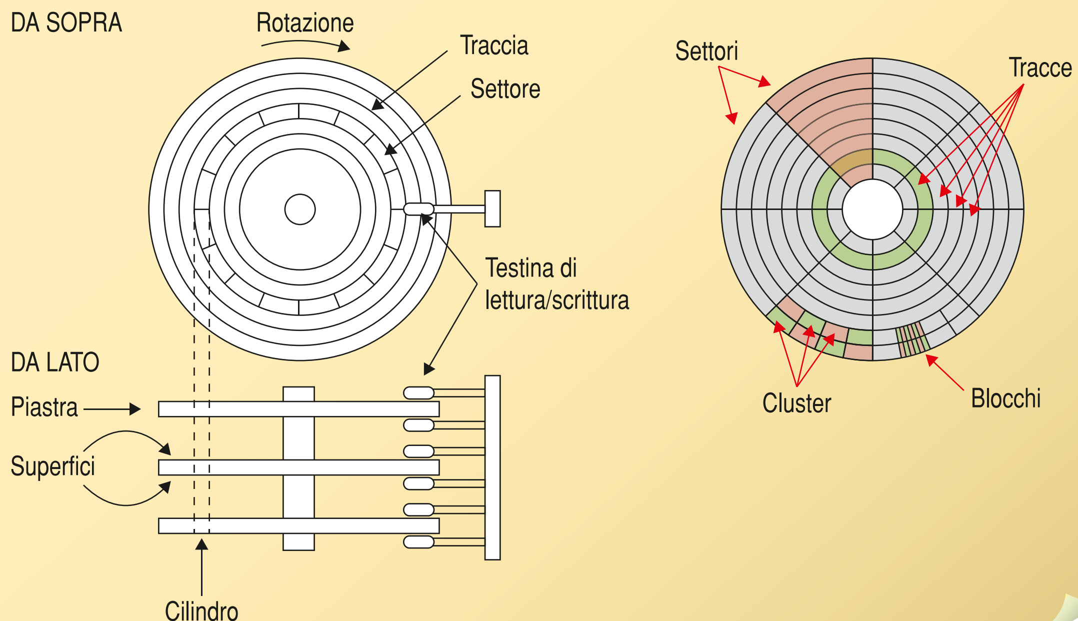
circa 1.6 GByte.

La seguente tabella mostra il contenuto dell'MBR:

Address			Description		Size in bytes
Hex	Oct	Dec			
0000	0000	0	Code area		440 (max 446)
01B8	0670	440	Disk signature (optional)		4
01BC	0674	444	Usually nulls; 0x0000		2
01BE	0676	446	Table of primary partitions (Four 16-byte entries, IBM partition table scheme)		64
01FE	0776	510	55h	MBR Signature	2
01FF	0777	511	AAh		
				MBR, total size: 446 + 64 + 2 =	512

Subito dopo il POST, viene generalmente chiamato l'interrupt 19h, che legge il boot sector del primo disco avviabile e lo carica in memoria a partire dalla locazione 7C00h. Quindi il registro SI viene inizializzato con il valore dell'indirizzo della prima locazione relativa alla riga della partition table che descrive la partizione dalla quale è stato caricato il boot record.
L'avvio di un sistema basato su architettura x86 è un processo piuttosto complesso, poiché inizialmente la CPU funziona in modalità reale, potendo accedere soltanto al primo MB di memoria il boot loader caricherà il kernel di Linux all'interno del primo mega di memoria RAM. Una volta avviato, il kernel Linux provvederà a far passare la CPU alla modalità protetta per avere a disposizione tutta la memoria fisicamente installata sul sistema.
Linux offre moltissimi bootloader diversi, i più noti probabilmente sono LILO (LInux LOader) e GRUB (GRand Unified Bootloader).

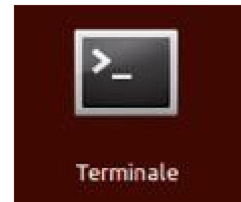
Riportiamo, per chiarezza, la struttura fisica di un disco:



ESEMPIO 1 *Leggiamo l'MBR da Linux Ubuntu*

La seguente procedura illustra come individuare i dati relativi all'MBR del disco usando l'ambiente di lavoro GUI fornito dal sistema operativo Linux, distribuzione Ubuntu.

1. Per prima cosa dobbiamo entrare nell'applicazione denominata Terminale, individuabile dalla seguente icona:



> L'ambiente Terminale verrà ampiamente utilizzato più avanti, in questo caso vogliamo solo utilizzarne alcuni comandi.

2. Nella finestra che appare, viene mostrato il prompt dei comandi di linea, che identifica l'utente principale del computer. Il comando seguente consente di visualizzare il dump dell'MBR:

```
sudo dd if=/dev/sda bs=512 count=1 | hexdump -c
```

Terminale

```
roberto@roberto: /
roberto@roberto:/$ sudo dd if=/dev/sda bs=512 count=1 | hexdump -c
```

3 Dopo aver digitato il comando e premuto Enter otteniamo la seguente schermata in cui abbiamo evidenziato le parti di nostro interesse:

```
roberto@roberto: ~
roberto@roberto:~$ sudo dd if=/dev/sda bs=512 count=1 | hexdump
[sudo] password for roberto:
0000000 63eb 1090 d08e 00bc b8b0 0000 d88e c08e
0000010 befb 7c00 00bf b906 0200 a4f3 21ea 0006
0000020 be00 07be 0438 0b75 c683 8110 fefe 7507
0000030 ebf3 b416 b002 bb01 7c00 80b2 748a 8b01
0000040 024c 13cd 00ea 007c eb00 00fe 0000 0000
0000050 0000 0000 0000 0000 0000 8000 0001 0000
0000060 0000 0000 faff 9090 c2f6 7480 f605 70c2
0000070 0274 80b2 79ea 007c 3100 8ec0 8ed8 bcd0
0000080 2000 a0fb 7c64 ff3c 0274 c288 bb52 0417
0000090 2780 7403 be06 7d88 17e8 be01 7c05 41b4
00000a0 aabb cd55 5a13 7252 813d 55fb 75aa 8337
00000b0 01e1 3274 c031 4489 4004 4488 89ff 0244
00000c0 04c7 0010 8b66 5c1e 667c 5c89 6608 1e8b
00000d0 7c60 8966 0c5c 44c7 0006 b470 cd42 7213
00000e0 bb05 7000 76eb 08b4 13cd 0d73 c2f6 0f80
00000f0 d084 be00 7d93 82e9 6600 b60f 88c6 ff64
0000100 6640 4489 0f04 d1b6 e2c1 8802 88e8 40f4
0000110 4489 0f08 c2b6 e8c0 6602 0489 a166 7c60
0000120 0966 75c0 664e 5ca1 667c d231 f766 8834
0000130 31d1 66d2 74f7 3b04 0844 377d c1fe c588
0000140 c030 e8c1 0802 88c1 5ad0 c688 00bb 8e70
0000150 31c3 b8db 0201 13cd 1e72 c38c 1e60 00b9
0000160 8e01 31db bff6 8000 c68e f3fc 1fa5 ff61
0000170 5a26 be7c 7d8e 03eb 9dbe e87d 0034 a2be
0000180 e87d 002e 18cd feeb 5247 4255 0020 6547
0000190 6d6f 4800 7261 2064 6944 6b73 5200 6165
00001a0 0064 4520 7272 726f 0a0d bb00 0001 0eb4
00001b0 10cd 3cac 7500 c3f4 2922 0004 0000 2080
00001c0 0021 fe83 ffff 0800 0000 f800 01df fe00
00001d0 ffff fe05 ffff 07fe 01e0 f002 001f 0000
00001e0 0000 0000 0000 0000 0000 0000 0000 0000
00001f0 0000 0000 0000 0000 0000 0000 0000 aa55
```

Area di codice

Firma del disco

Byte null

Tabella delle partizioni primarie

AA 55h indica la fine dell'MBR

Area di codice

Possiamo vedere come in questo esempio l'MBR contenga effettivamente del codice, in grado di caricare un sistema operativo. In questo esempio non vogliamo studiare come lavora effettivamente l'MBP, ma scoprirne soltanto la struttura.

Firma del disco

Contiene l'ID del produttore del disco.

Byte Null

Si tratta di 2 byte posti a zero per separare l'MBP dalla tabella delle partizioni.

Tabella delle partizioni primarie

In questa sezione vi sono 64 bytes che descrivono le partizioni primarie. Ogni 16 byte vi è la descrizione di una singola partizione. Anche se non conosciamo nel dettaglio il significato dei valori esadecimali indicati, possiamo comunque affermare che in questa tabella delle partizioni vi sono due partizioni primarie descritte da questo MBR, in quanto nelle rimanenti due righe i valori sono tutti a zero.

Fine dell'MBR

La sequenza 55AAh definisce la fine del MBR, in tal modo il sistema riconosce il termine dei 512 bytes del Master Boot Record.

Vediamo di comprendere il significato delle righe della tabella delle partizioni. Innanzi tutto teniamo presente che i valori sono da leggere a 2 a 2 in quanto trattasi di Word. La sequenza della partizione 1 risulta pertanto:

80 20 21 00 83 FE FF FF 00 08 00 00 00 F8 DF 01

80 indica lo stato della partizione, 80 per primaria avviabile, 00 per non avviabile. In questo caso 80 indica che è avviabile. Possiamo notare che la seconda partizione non è avviabile.
I 3 byte successivi indicano l'indirizzo assoluto dell'inizio della partizione nell'ordine: la testina, il settore e il cilindro. (20 21 00)
Il byte successivo è relativo al file system presente. In questo caso è 83 e indica un sistema Linux.
I 3 byte successivi indicano l'indirizzo assoluto dell'ultimo settore della partizione nell'ordine: la testina, il settore e il cilindro. (FE FF FF)
I 4 byte successivi indicano il numero di blocchi prima della partizione. (00 08 00 00)
I 4 byte successivi indicano il numero di blocchi che costituiscono la partizione. (00 F8 DF 01)

Il seguente comando ci consente di confermare quanto detto sopra:

```
sudo dd if=/dev/sda bs=512 cont=1 of=mbr.bin
```

Il comando trascrive il contenuto dell'MBR in un file binario, rendendo più agevole la lettura delle informazioni che riguardano le partizioni. Infatti mediante il comando successivo:

```
file mbr.bin
```

stampiamo a video il contenuto del file mbr.bin. Possiamo notare che vi sono due partizioni nel sistema, la prima inizia dalla testina 32, settore 2048, occupando 31455232 settori, mentre la seconda inizia alla testina 254, settore 31459326, occupando 2093058 settori. Possiamo inoltre notare che l'ID della prima partizione è 83h (Linux), mentre la seconda è 05h (partizione estesa):

```
roberto@roberto: /
roberto@roberto:/$ sudo dd if=/dev/sda bs=512 count=1 of=mbr.bin
1+0 record dentro
1+0 record fuori
512 byte (512 B) copiati, 4,1612e-05 s, 12,3 MB/s
roberto@roberto:/$ file mbr.bin
mbr.bin: x86 boot sector; partition 1: ID=0x83, active, starthead 32, startsector 2048, 31455232 secto
rs; partition 2: ID=0x5, starthead 254, startsector 31459326, 2093058 sectors, code offset 0x63
roberto@roberto:/$
```

Il Bootloader GRUB

Il boot loader più diffuso tra i sistemi Linux è senza dubbio GRUB (GRand Unified Bootloader), che viene prodotto nel 1995 da E. Boleyn, ed è utilizzato in tutte le più recenti distribuzioni di Linux. Le sue caratteristiche principali sono le seguenti:

- supporta l'avvio di distribuzioni di S.O. aderenti alle specifiche Multiboot (GNU/Linux, FreeBSD, NetBSD, OpenBSD);
- supporta l'avvio in ◀ chain-loading ▶ di sistemi operativi non-Multiboot, come per esempio MSDOS, Windows, OS/2;
- supporta numerosi file system non Linux;
- consente di gestire l'avvio del sistema operativo desiderato attraverso un'interfaccia utente con un menu.



◀ Chain loading È una tecnica usata per passare il controllo dal boot manager al boot sector: il settore di boot viene caricato dal disco sostituendo il boot sector presente in memoria da cui il boot manager stesso verrà caricato ed eseguito. ▶

All'avvio, GRUB visualizza un elenco dei possibili sistemi operativi avviabili, come si vede nella figura di esempio:

```
GNU GRUB  version 1.98-1ubuntu7

Ubuntu, with Linux 2.6.32-28-generic
Ubuntu, with Linux 2.6.32-28-generic (recovery mode)
Ubuntu, with Linux 2.6.32-24-generic
Ubuntu, with Linux 2.6.32-24-generic (recovery mode)
Ubuntu, with Linux 2.6.31-11-rt
Ubuntu, with Linux 2.6.31-11-rt (recovery mode)
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)

   Use the ↑ and ↓ keys to select which entry is highlighted.
   Press enter to boot the selected OS, 'e' to edit the commands before booting or 'c' for a command-line.
```

Gli stadi di avvio

GRUB è formato da diversi stadi di avvio, chiamati stage, ciascuno identificato da un file. Due stage sono essenziali e sono presenti nei file stage1 e stage2, gli altri stage sono opzionali. Lo stage1 è il file che contiene le informazioni di avvio, si trova infatti nell'MBR o nel boot sector di una partizione. Poiché un boot sector occupa 512 byte, la dimensione del file ricalca tale grandezza, quindi esattamente 1/2 kB. Il file stage1 fa sì che venga caricato in memoria il file stage2 prelevandolo dal disco. Il file stage2 rappresenta il nucleo di GRUB, esso infatti risiede sul file system e permette di scegliere il sistema di avvio.

Installazione di GRUB

Per utilizzare GRUB come boot loader del sistema dobbiamo innanzi tutto installare sul sistema il relativo pacchetto contenente appunto i file necessari alla configurazione e installazione di GRUB come boot loader. Per installare il pacchetto GRUB dobbiamo usare il seguente comando:

```
sudo apt-get install grub
```

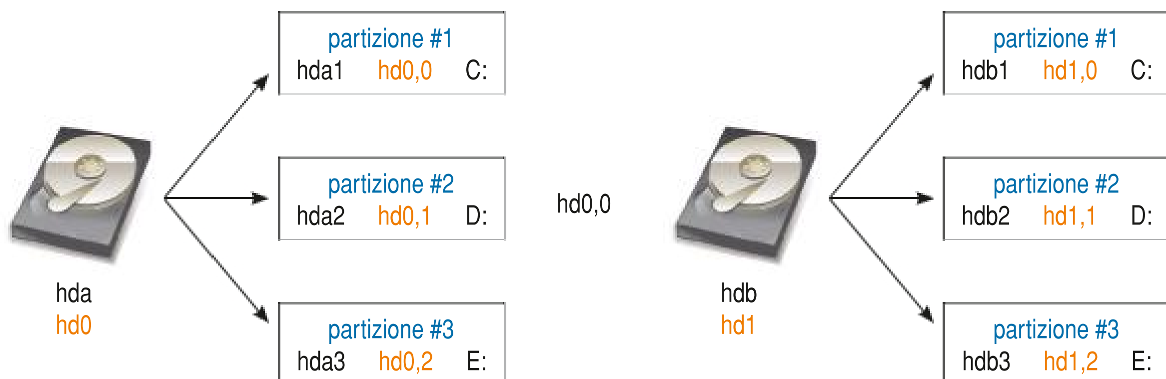
```
roberto@roberto:~$ sudo apt-get install grub
[sudo] password for roberto:
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
Pacchetti suggeriti:
  grub-legacy-doc mdadm
I seguenti pacchetti saranno RIMOSSI:
  grub-gfxpayload-lists grub-pc grub2-common
I seguenti pacchetti NUOVI saranno installati:
  grub
0 aggiornati, 1 installati, 3 da rimuovere e 191 non aggiornati.
È necessario scaricare 330 kB di archivi.
Dopo quest'operazione, verranno occupati 78,8 kB di spazio su disco.
Continuare [S/n]?
```

Digitando “s” viene installato il pacchetto. Prima di vederne il funzionamento dobbiamo imparare come vengono denominati i dischi e le partizioni dai S.O.

- Windows assegna a ogni partizione collocata su un qualsiasi disco un valore simbolico composto da una lettera progressiva e dai due punti, chiamata unità logica. Per esempio A: e B: per i dischi floppy, C: e seguenti per dischi fissi e altre unità.
- Unix assegna un nome al disco nella forma hda, hdb, hdc ecc. e per ogni disco definisce un valore fisso e progressivo che identifica ciascuna delle partizioni (es: hda1, hda5, hdb3). I numeri da 1 a 4 sono riservati alle quattro possibili partizioni primarie, mentre si usano valori superiori a 5 per denominare le partizioni logiche.
- GRUB utilizza una propria notazione, diversa dalle due precedenti, che assegna a ogni disco fisso il prefisso hd seguito dall’identificativo del disco e della partizione:

hd(<numero_disco>,<numero_partizione>)

La figura seguente mostra come vengono individuati due Hard disk, con le relative partizioni:



ESEMPIO 2 *Modificare il file GRUB*

Vediamo come modificare il menu di avvio che consente di definire eventuali altri S.O. presenti su altre partizioni. Per fare questo digitiamo il comando che apre il file di configurazione di GRUB, che contiene le informazioni sulle partizioni e sulle versioni di S.O. inserite:

```
sudo gedit /etc/default/grub
```

```
*grub (/etc/default) - gedit
File Modifica Visualizza Cerca Strumenti Documenti Aiuto
Apri  Salva  Annulla
*grub ×
# If you change this file, run 'update-grub' afterwards to
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo De
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

title GNU Linux
root (hd0,1)
kernel /boot/vmlinuz root=/dev/sda1 vga=ext
initrd /boot/initrd
savedefault

title Windows XP
root(hd0,1)
savedefault
maheactive
chainloader +1
```

Innanzi tutto vediamo due comandi assai importanti:
- GRUB_DEFAULT=0, indica che la prima voce del menu è quella di default, cioè la prima voce del menu di avvio. Possiamo individuare le voci del menu di avvio in quanto contraddistinte dalla voce title accanto.
- GRUB_TIMEOUT=10, indica che l'avvio sarà automaticamente eseguito dalla partizione di default, trascorsi 10 secondi senza aver effettuato altre scelte.

Ogni nuova voce del menu a tendina che apparirà all'avvio è data dal comando title, seguita dal testo che indica il titolo che viene visualizzato. La partizione su cui è installato il sistema operativo che verrà avviato selezionando il testo indicato da title è identificata dal comando root seguito da un identificativo di una partizione secondo la notazione grub vista in precedenza. In questo esempio possiamo notare due partizioni avviabili, una per Linux e una per Windows XP.

L'avvio del sistema operativo

Il boot loader ha il compito di caricare in memoria il kernel, decomprimerlo, mentre la CPU è in stato di protected mode, e di avviarlo passandogli eventuali parametri. Il kernel si preoccupa di compiere le operazioni necessarie all'avvio dell'intero sistema operativo, ovvero dei processi fondamentali per il suo utilizzo. L'unico processo che viene avviato dal kernel è init, tramite l'esecuzione del file /sbin/init, il quale poi si preoccupa di avviare gli altri processi in base al file /etc/inittab che contiene le indicazioni dei processi da avviare. Init esegue l'avvio dei processi che si occupano della gestione del sistema e dei terminali, chiamati demoni (◀ daemon ▶).



◀ Daemon Si tratta di programmi eseguiti in background e residenti, che gestiscono le richieste fatte al S.O., come per esempio quelli relativi alla stampa, alla comunicazione con le porte USB, oppure per la schedulazione dei lavori. ▶

Come detto il primo file eseguito da init è inittab che si occupa di eseguire le seguenti operazioni:

- montare il file system /proc;
- impostare l'ora dell'orologio del sistema (clock) a quella dell'orologio del BIOS;
- impostare il layout della tastiera (keymap) e i caratteri del sistema (font);
- attivare il funzionamento dell'interfaccia di rete;
- inizializzare il controller USB (Universal Serial Bus);
- eseguire un controllo di integrità sul file system nel caso si accorga che il precedente spegnimento del sistema sia avvenuto in modo non corretto;
- inizializzare i dispositivi Plug'N'Play;
- inizializzare il LVM (Logical Volume Management);
- attivare lo swap;
- avviare i dispositivi RAID;
- inizializzare il LVM per i dispositivi RAID;
- montare le altre partizioni;
- inizializzare le porte seriali e Firewire (se rilevate).

Un sistema, oltre a eseguire i comandi impartiti dagli utenti, può mettere a disposizione degli stessi alcuni servizi come la posta elettronica, il server web, la gestione centralizzata degli utenti. La procedura di inizializzazione del sistema ha il compito di avviare il sistema operativo e di fermarlo, attivando e disattivando tutti i servizi necessari, intervenendo nell'avvio e nella terminazione dei processi a essi relativi. Tali processi, chiamati appunto daemon, forniscono i servizi desiderati e hanno la caratteristica di essere avviati automaticamente all'avvio del sistema e rimanere in esecuzione in background finché il sistema non viene spento.

La procedura di avvio dei daemon è un sistema automatico per l'avvio dei servizi che si desiderano attivare subito dopo l'avvio del sistema e terminare subito prima dell'arresto del sistema stesso, derivata dalla procedura di avvio del S.O. Unix.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 I sistemi operativi vengono installati:

a) nell'MBR
b) in una partizione
c) in un file speciale
d) nel kernel

2 I sistemi operativi che gestiscono attività concorrenti sono chiamati:

a) sistemi operativi concorrenti
b) sistemi operativi multitasking
c) sistemi operativi processing
d) sistemi operativi multiutenti

3 La fase di avvio di un S.O. si chiama:

a) BIOS
b) POST
c) bootstrap
d) kernel

4 Il programma che sovrintende le operazioni di caricamento di un S.O. si chiama:

a) boot loader
b) bootstrap
c) BIOS
d) POST

5 Metti in ordine cronologico numerandole da 1 a 4 le seguenti fasi di avvio di un sistema:

a) .... il bootloader carica l'immagine del kernel
b) .... iinizia la fase vera e propria di caricamento
c) .... il BIOS carica il boot loader
d) .... il boot loader indica alla CPU di eseguire il kernel

6 Durante la fase di accensione del sistema la CPU esegue:

a) un Jump condizionato, se non ci sono errori del sistema, all'indirizzo FFFF0h per eseguire il BIOS
b) un Jump all'indirizzo 0FFFFh per eseguire il BIOS
c) un Jump all'indirizzo FFFF0h per eseguire il BIOS
d) un Jump all'indirizzo FFFF0h per eseguire il bootstrap

7 Quanti beep vengono segnalati in caso di errore del processore?

a) 3 beep
b) 1 beep lungo
c) 5 beep
d) 6 beep

8 Metti in ordine cronologico, numerandole da 1 a 4, le seguenti operazioni effettuate dal boot loader:

a) .... Viene caricato in memoria, a partire dall'indirizzo 7C00h, il primo settore del primo disco avviabile (MBR).
b) .... Viene eseguito il codice chiamato MBP.
c) .... Viene verificato se il dispositivo è avviabile.
d) .... L'SBP avvia il processo di caricamento del sistema Linux avviandone il kernel.
e) .... Viene caricata la seconda sezione del boot loader, chiamata SBP.

9 La Master Boot Table è una tabella che contiene:

a) informazioni sulle partizioni
b) informazioni sull'MBR
c) l'indirizzo del primo cluster libero di ogni partizione
d) informazioni sul disco fisso

10 L'MBR è il primo settore di un disco, posizionato sul:

a) cilindro 0, testina 0, e settore 0
b) cilindro 1, testina 0, e settore 0
c) cilindro 0, testina 0, e settore 1
d) cilindro 1, testina 1, e settore 0

11 Cosa contiene l'MBR?

a) L'MBP e la partition table
b) L'EBP e l'EBT
c) L'MBT e l'SBT
d) L'MBP e l'EBP

12 Quali tra i seguenti sono bootloader di Linux?

a) Ubuntu
b) LILO
c) Greg
d) GRUB

13 Quale byte nel MBR indica che una partizione è avviabile?

a) 00 primaria avviabile, 01 non avviabile
b) 80 primaria avviabile, 81 non avviabile
c) 00 primaria avviabile, 80 non avviabile
d) 80 primaria avviabile, 00 non avviabile

14 Come identifica gli hard disk GRUB?

a) sd numero partizione numero disco
b) hd numero disco numero partizione
c) hd numero partizione numero disco
d) sd numero disco numero partizione

Test vero/falso

1 Una partizione può contenere più sistemi operativi. V F
2 Un sistema operativo avviabile può essere installato solo in una partizione estesa. V F
3 Una partizione estesa differisce da una primaria perché a sua volta può essere ulteriormente partizionata. V F
4 Linux può essere installato solo in una partizione estesa. V F
5 Il programma che sovrintende l'operazione di caricamento viene chiamato boot loader. V F
6 La sequenza di boot è contenuta nel BIOS. V F
7 Il BIOS dopo aver effettuato il bootstrap effettua il POST. V F
8 Nel riavvio a caldo non viene effettuato il POST. V F
9 Le routine del BIOS che controllano il circuito di interrupt riconoscono gli hard disk e i dispositivi P&P. V F
10 L'MBR è un programma che risiede all'interno dell'MBP. V F
11 L'EBR è un MBR relativo a ciascuna partizione estesa. V F
12 L'EBT contiene il settore di inizio della partizione estesa. V F
13 La dimensione di un disco è data da: (numero di cilindri) × (numero di testine) × (63 settori/traccia) × (512 byte/settore). V F
14 All'avvio di un sistema x86 inizialmente la CPU funziona in modalità protetta. V F
15 Le partizioni primarie possono essere solo 4 al massimo. V F
16 La dimensione dell'MBR è di 512 bytes. V F
17 La sequenza 00AAh definisce la fine del MBR. V F
18 La tecnica del chain loading passa il controllo dal boot manager al boot sector. V F

LEZIONE 2

IL FILE SYSTEM DI LINUX

IN QUESTA LEZIONE IMPAREREMO...

- a conoscere la struttura fisica e logica del file system Linux
- a definire la struttura dell'FHS
- a individuare le principali distribuzioni Linux
- a conoscere la struttura logica del file system Linux

La storia del sistema operativo GNU/Linux

Innanzi tutto affermiamo che Linux non è un sistema operativo, bensì un kernel del S.O. Infatti il sistema operativo vero e proprio si chiama GNU/Linux. Occorre capire da dove deriva per comprendere quanto affermato.
Inizialmente il software era condiviso, infatti ognuno scriveva il software per le proprie necessità, come per esempio per le schede perforate, ed era liberamente modificabile e migliorabile da chiunque. Tuttavia dal 1976 le cose cambiarono quando Bill Gates si fece portavoce dell"introduzione del software proprietario, definiva infatti impensabile la produzione di software di buona qualità senza che gli investimenti fatti per la sua produzione non fossero poi protetti dalla proprietà intellettuale e da contratti di licenza restrittivi.
Richard Stallman, un hacker del Massachussets Institute of Technology, si rese conto che la chiusura delle licenze del software avrebbe impedito la collaborazione tra programmatori, e incomincia a lavorare al progetto ◀ GNU ▶.
Il progetto GNU viene avviato da Richard Stallman nel 1983. Il principale obiettivo del progetto è la realizzazione di un sistema operativo completamente libero e aperto, ispirato a Unix, in cui ogni utente o programmatore potesse accedere al codice sorgente del software, per poterlo studiare e migliorare. Nasce tuttavia un problema su come poter garantire queste libertà in accordo con la legislatura vigente: per fare questo fondò, insieme ad alcuni legali una cosiddetta licenza virale, chiamata GPL (GNU Public License). Si tratta del primo esempio di licenza OpenSource, interamente basata sul concetto di Copyleft.



◀ GNU Nel progetto denominato GNU/Hurd, dove Hurd è il kernel, Stallman riscrive il codice sorgente del sistema operativo Unix. Infatti il termine GNU è un acronimo ricorsivo di Not Unix. ▶

I punti salienti di questa licenza sono proprio la liberta di utilizzare, modificare, copiare e redistribuire il software che deve essere redistribuito con una licenza analoga a quella con la quale è stato rilasciato.
Nel 1990 il progetto è quasi completo, manca solo il nucleo del S.O., il kernel viene sviluppato da uno studente dell'Università di Helsinki, Linus Torvalds, che mise a punto un kernel chiamato Freax. Venne rilasciata una versione con licenza GPL, memorizzata in una cartella condivisa sul server dell'Università di Helsinki. La directory era di proprietà di un amico di Torvalds che la chiamò Linux in onore dell'amico. Fu a quel punto che tutti iniziarono a chiamare il kernel con quel nome: Linux.

> Cosa garantisce una licenza open source come la GPL.
> - Libertà di eseguire il programma, come desideri.
> - Libertà di studiarne il codice sorgente e modificarlo.
> - Libertà di fare copie e distribuirle agli altri.
> - Libertà di pubblicare versioni modificate.

Da quel momento nacquero numerose versioni chiamate ◀ distribuzioni ▶.



◀ Distribuzioni Siccome non esiste una azienda incaricata di distribuire Linux, il compito di realizzare i programmi d'installazione e fornire le applicazioni di uso più comune è stato assunto da aziende, università, gruppi di utenti e persino singoli individui. Il SO Linux e tutti gli applicativi creati vengono indicati con il nome distribuzione o versione. Le distribuzioni si compongono di due elementi di base:
- kernel (programma che ha il compito di gestire il funzionamento dei componenti del sistema e assicurare la corretta installazione degli applicativi);
- X Window (programma responsabile dell'interfaccia grafica del sistema operativo). ▶

Una distribuzione non è altro che un insieme di software selezionato per uno specifico scopo, scelto in modo tale, solitamente, che possa essere installato e utilizzato nell'ambito per il quale è stato pensato. Vengono infatti realizzate solitamente per scopi ben precisi, come per esempio desktop, server, grafica, streaming, firewall, liveCD, musica. Vediamo quali sono i componenti che le caratterizzano:
- l'interfaccia grafica (GNOME, KDE, XFCE, Enlightenemnt, OpenBox, FluxBox...);
- il sistema di pacchettizzazione del software (.deb, .RPM, URPMI, SRC...);
- le Community.

Vediamo alcune distribuzioni più note:

Debian



È una delle distribuzioni più storiche, amata da molti programmatori e sistemisti proprio per la garanzia di estrema stabilità. Esiste per numerose architetture (Alpha, AMD64, ARM, HP PA-RISC, Intel x86, Intel IA-64, MIPS, PowerPC, IBM S/390, SPARC), tuttavia vengono rilasciati pochi aggiornamenti. Possiede come package .deb e come kernel Linux oppure FreeBSD.

redhat

Si tratta di una distribuzione a pagamento per utilizzi prettamente aziendali. Il costo è legato solo ai CD che contengono i file binari e a una eventuale assistenza. Tuttavia è la distribuzione probabilmente meglio riuscita ed è soprattutto utilizzata nella sua versione server. È compatibile con le architetture IA-32 e x86-64. Il gestore di pacchetti è .RPM mentre il kernel è di tipo Linux.



Knoppix



Nasce come versione LiveCD, cioè avviabile da CDROM, tuttavia ha avuto un certo successo anche nella versione tradizionale. È di derivazione Debian. Possiede una gestione dei pacchetti particolare, scaricabile da Internet attraverso un file system chiamato UnionFS. È compatibile con le architetture i386 e x86-64. Il kernel è di tipo Linux.

Fedora



È una distribuzione assai innovativa, nasce nel 2004 ed è molto sicura, possiede molto software. È prioritariamente orientata alle versioni server e non desktop, possiede come gestore dei package .rpm e come kernel Linux. È adatta a diverse architetture (IA-32, x86-64, PowerPC).

Ubuntu



Prevede un aggiornamento ogni 6 mesi ed è molto stabile e affidabile. È stato uno dei primi a possedere una interfaccia grafica in stile Windows. Utilizza come gestore dei package .deb ed è compatibile con le architetture x86 e MAD64, mentre il kernel è Linux.

Il file system

Uno dei concetti fondamentali dell'architettura di Linux è il cosiddetto everything is a file, cioè l'accesso ai vari dispositivi di I/O viene effettuato attraverso un'interfaccia astratta che tratta le periferiche come se fossero dei normali file di dati. In tal modo possiamo accedere a qualsiasi periferica del computer (per esempio i dischi) mediante i cosiddetti ◀ device file ▶.



◀ Device file La traduzione dall'inglese è: file dispositivo. I programmi possono leggere, scrivere e compiere operazioni direttamente sulle periferiche, agendo su questi file speciali, usando le stesse funzioni che si usano per i normali file di dati. I file di dispositivo sono contenuti nella directory /dev (/dev/hda, /dev/sda…). I driver sono opportuni programmi che effettuano l'associazione tra i dispositivi e i relativi file di dispositivo: il driver ATA associa i dispositivi collegati a tale bus ai file di dispositivo /dev/hd*, mentre quello relativo al bus SCSI associa i dispositivi di quel tipo ai file /dev/sd*. ▶

La seguente tabella mostra i nomi dei device file e il relativo significato:

File	Descrizione
/dev/hdx[n]	Disco ATA x o n-esima partizione del disco
/dev/sdx[n]	Disco SCSI x (o SATA, USB, CD/DVD) o n-esima partizione di questi tipi di dischi
/dev/edx[n	Disco ESDI (obsoleto) x o n-esima partizione di questi tipi di dischi
/dev/hdx[n]	Disco XT (obsoleto) x o n-esima partizione di questi tipi di dischi
/dev/fdn	Floppy disk n

Per poter accedere a questi tipi particolari di file il kernel ci mette a disposizione opportune interfacce che consentono di leggerne il contenuto, provvedendo in tal modo a organizzare e rendere accessibile in maniera opportuna l'informazione tenuta sullo spazio grezzo disponibile sui dischi. Questo viene fatto strutturando l'informazione sul disco attraverso quello che si chiama un file system, essa poi viene resa disponibile ai processi attraverso quello che viene chiamato il montaggio (mount) del file system stesso.

La gestione dei pacchetti

I pacchetti in Linux vengono gestiti da un programma chiamato packet manager che ha il compito di automatizzare il processo di installazione, aggiornamento, configurazione e rimozione dei pacchetti software.
Il packet manager o gestore dei pacchetti, viene utilizzato per installare, aggiornare, verificare e rimuovere software del sistema operativo in maniera intuitiva e aiuta a risolvere le dipendenze tra i pacchetti.
Linux, analogamente a Windows, distribuisce il software sotto forma di pacchetti incapsulati tuttavia in un singolo file. I pacchetti spesso includono anche altre importanti informazioni, come il

nome completo, la versione, e il fornitore del software, e una lista di altri pacchetti, conosciuti come dipendenze, a volte necessarie al software per funzionare correttamente.
I packet manager per mantenere l'usabilità dei pacchetti adottano diverse tecniche:
- verificano il checksum dei file per evitare differenze tra le versioni locali e ufficiali di un pacchetto;
- possiedono strumenti per l'installazione, l'aggiornamento, e la rimozione;
- gestiscono le dipendenze per la distribuzione del software funzionante da un pacchetto;
- controllano gli aggiornamenti per fornire le ultime versioni dei software, che spesso includono riparazioni di difetti e aggiornamenti nel campo della sicurezza.

Vediamo alcuni sistemi di gestione di pacchetti realizzati da sistemi operativi:
- Advanced Packaging Tool, conosciuto anche come APT, è lo strumento che gestisce i pacchetti in formato .deb;
- RPM Package Manager, è il gestore di pacchetti RPM. Introdotto da Red Hat, viene oggi utilizzato da molte altre distribuzioni Linux. RPM è il formato base standard, insieme a deb di Debian, per la pacchettizzazione di Linux.

Le partizioni

Per poter essere utilizzato, un disco va innanzitutto preparato mediante l'operazione di partizionamento, che consiste nella suddivisione dello spazio totale del supporto in sottoinsiemi più piccoli, detti appunto partizioni.
A seconda del file system utilizzato, il partizionamento può migliorare le prestazioni dello stesso oppure limitare lo spazio sprecato, infatti lo spazio di memoria a disco disponibile dipende in larga misura dalla dimensione dei blocchi del file system.
Il partizionamento del disco viene effettuato per mezzo del comando fdisk, oppure del comando parted che consente anche il ridimensionamento di alcuni tipi di partizioni già esistenti.

L'esigenza del partizionamento è nata dal fatto che i BIOS e i sistemi operativi meno recenti non riuscivano a gestire dischi di grandi dimensioni.

ESEMPIO 3 *Visualizzare le partizioni e modificarle*

Attraverso il comando fdisk -l possiamo conoscere quali partizioni esistono. Digitiamo:

```
sudo fdisk -l
```

Otteniamo la seguente schermata in cui si possono individuare le partizioni.

```
roberto@roberto: ~
roberto@roberto:~$ sudo fdisk -l

Disk /dev/sda: 17.2 GB, 17179869184 bytes
255 testine, 63 settori/tracce, 2088 cilindri, totale 33554432 settori
Unità = settori di 1 * 512 = 512 byte
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Identificativo disco: 0x00042922

Dispositivo Boot      Start         End      Blocks   Id  System
/dev/sda1   *          2048    31457279    15727616   83  Linux
/dev/sda2          31459326    33552383     1046529    5  Esteso
/dev/sda5          31459328    33552383     1046528   82  Linux swap / Solaris
roberto@roberto:~$
```

Come possiamo vedere esiste anche una partizione particolare chiamata ◀ Linux swap ▶, che vedremo nella prossima lezione, essere indispensabile per il funzionamento del S.O.



◀ Linux swap Il S.O. Ubuntu per poter lavorare deve avere a disposizione minimo due partizioni. Una per il sistema operativo e per i dati e l'altra per lo swap. Lo swap è una partizione di dimensione almeno doppia rispetto alla memoria RAM se quest'ultima è inferiore a 1 GB. Si tratta di una memoria di servizio che viene utilizzata in aiuto alla RAM quando questa è insufficiente. Quando il computer è in stato di ibernazione o sospensione viene usata per memorizzare lo stato della RAM. ▶

Per modificare le partizioni, per esempio aggiungendone un'altra dobbiamo utilizzare il comando parted. Digitiamo il seguente comando:

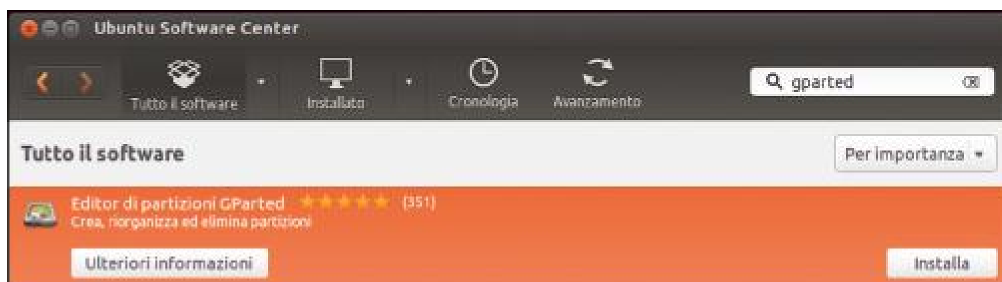
```
sudo parted
```

Come possiamo notare appare una schermata in cui è possibile digitare un comando per creare o modificare, oppure ancora eliminare una partizione esistente:

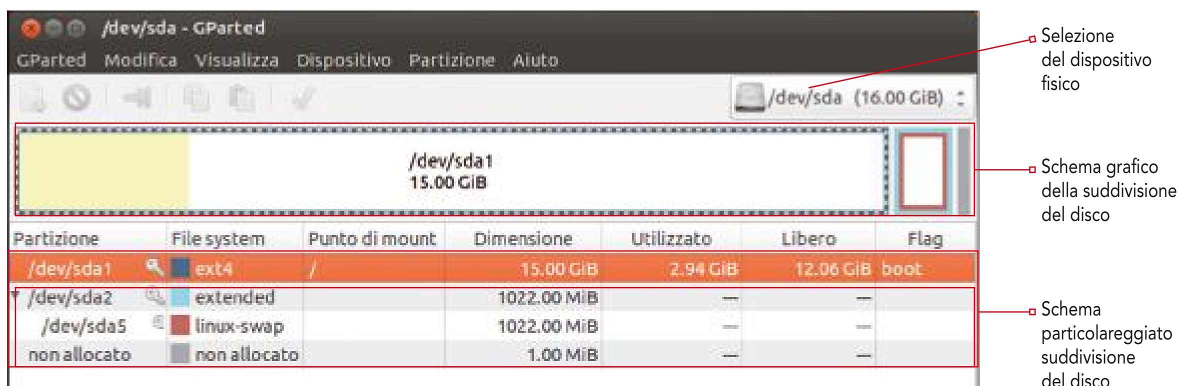
```
roberto@roberto:~$ sudo parted
[sudo] password for roberto:
GNU Parted 2.3
Viene usato /dev/sda
Benvenuti in GNU Parted. Digitare "help" per l'elenco dei comandi.
(parted) help
  align-check TIPO N                       Controlla la partizione N per
        l'allineamento TIPO(min|ott)
  check NUMERO                            Esegue un semplice controllo sul file
        system
  cp [DEVICE_INZIALE] DAL_NUMERO AL_NUMERO Copia un file system in un'altra
        partizione
  help [COMANDO]                          Mostra l'aiuto generale o sul COMANDO
  mklabel,mktable TIPO_ETIC               Crea una nuova etichetta del disco
        (tabella delle partizioni)
  mkfs NUMERO TIPO_FS                     Crea un file system TIPO_FS sulla
        partizione NUMERO
  mkpart TIPO_PART [TIPO_FS] INIZIO FINE  Crea una partizione
  mkpartfs TIPO_PART TIPO_FS INIZIO FINE  Crea una partizione con un file
        system
  move NUMERO INIZIO FINE                 Sposta la partizione NUMERO
  name NUMERO NOME                        Chiama la partizione NUMERO come NOME
  print [device|free|list,all|NUMERO]     Visualizza la tabella delle
        partizioni, i device disponibili, lo spazio libero, tutte le partizioni
        trovate o una particolare partizione
  quit                                    Esce dal programma
  rescue INIZIO FINE                      Ripristina una partizione persa
        vicino a INIZIO e FINE
  resize NUMERO INIZIO FINE               Ridimensiona la partizione NUMERO e
        il suo file system
  rm NUMERO                               Elimina la partizione NUMERO
  select DEVICE                           Sceglie il device da modificare
```

Esiste tuttavia un pacchetto assai più evoluto, chiamato gparted, che può essere installato su una chiavetta USB oppure su di un disco ottico. Si tratta di un utility molto efficiente e di utilizzo GUI che permette di creare e modificare partizioni esistenti in modo assai più agevole. La procedura seguente mostra come utilizzare questa utility con Ubuntu:

1. Per installarlo occorre entrare nell'applicazione Software Center, quindi digitare il nome del software che si intende scaricare dal menu di ricerca.



2. A questo punto, il programma appare nella scheda laterale, e può essere eseguito. Una volta eseguito ci mostra le partizioni disponibili e l'eventuale spazio non partizionato. In questo caso possiamo notare che è rimasto 1 MB di spazio non partizionato.



3. Dopo esserci posizionati sullo spazio da partizionare facciamo click sul menu Partizione e scegliamo Nuova, appare la seguente finestra:



4 dopo aver assegnato un nome alla partizione (Ubuntu2) confermiamo con Aggiungi. Otteniamo la seguente partizione:



5 Per confermare le operazioni eseguite dobbiamo fare click sul segno di spunta di colore verde. Appare la seguente finestra di conferma, facciamo click su Applica:



6 Adesso possiamo verificare l'effettiva presenza della partizione con il comando fdisk -l:

```
roberto@roberto:~$ sudo fdisk -l
[sudo] password for roberto:

Disk /dev/sda: 17.2 GB, 17179869184 bytes
255 testine, 63 settori/tracce, 2088 cilindri, totale 33554432 settori
Unità = settori di 1 * 512 = 512 byte
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Identificativo disco: 0x00042922

Dispositivo Boot      Start         End      Blocks   Id  System
/dev/sda1   *          2048    31457279    15727616   83  Linux
/dev/sda2          31459326    33552383     1046529    5  Esteso
/dev/sda3          33552384    33554431        1024   83  Linux
/dev/sda5          31459328    33552383     1046528   82  Linux swap / Solaris
```

Come possiamo notare la partizione è stata creata ed è la terza nell'elenco, quella cioè indicata dall'Id 83 e da 1024 blocchi, che formano 1 MB di dimensione totale.



Zoom su...

PARTIZIONI

In Linux possiamo modificare la dimensione delle partizioni, riducendole, aumentandole, eliminandole o spostandole su aree diverse. Queste operazioni sono spesso distruttive a meno che non si usino dei particolari software (gparted) con la capacità di modificare la dimensione delle partizioni conservando i dati in esse contenuti e la struttura del file system che le caratterizzano.
Le partizioni possono essere:

1. partizioni primarie;
2. partizioni estese;
3. partizioni logiche.

Le partizioni primarie sono quelle partizioni direttamente puntate da un record della tabella delle partizioni dell'MBR.
Le partizioni estese sono in invece puntatori a tabelle di partizioni. Se infatti il codice presente nel byte con offset 4 del record della tabella delle partizioni dell'MBR indica una partizione estesa (05h per esempio) tale record non punta a una partizione vera e propria bensì a una ulteriore tabella di partizioni che conterrà, a sua volta, un insieme di record ognuno contenente i dati relativi a partizioni così dette logiche. Con questo meccanismo si ha la possibilità di superare il limite delle 4 partizioni altrimenti puntate dalla sola tabella dell'MBR.

Directory e montaggio

In Linux, a differenza di quanto avviene in altri sistemi operativi, tutti i file vengono tenuti all'interno di un albero la cui radice (root directory) viene montata all'avvio, ciascun file viene identificato dal proprio percorso (pathname), composto da una serie di nomi separati dallo slash ("/").
Dopo la fase di inizializzazione il kernel riceve dal boot loader il file system da montare (mount): rappresenterà la radice dell'albero (/). Anche una directory viene vista dal sistema come file, il kernel la riconosce come directory e il suo scopo è quello di contenere file o altre directory secondo no schema ad albero. Il termine montare (mount) deriva dal comando relativo che permette di collegare un dispositivo o una partizione in un determinato percorso relativo all'albero delle cartelle.
Il comando opposto è umount che disconnette un volume. Per montare un determinato dispositivo all'interno di una directory usiamo il comando mount, la cui sintassi è indicata di seguito:

```
mount -t FILE_SYSTEM /percorso/dispositivo /percorso/di/montaggio
```

dove al posto di FILE_SYSTEM va inserito il tipo di file system con cui è stato formattato. Se si vuole per esempio montare in /mount/disco una partizione ext4 localizzata in /dev/sda2, il comando da lanciare è il seguente:

```
mount -t ext4 /dev/sda2 /mount/disco
```

Per rimuovere, quindi smontare, un dispositivo dobbiamo usare il comando umount e specificare il device oppure la directory sulla quale il dispositivo è stato precedente montato. Ad esempio se specifichiamo il device:

```
sudo umount "device"
```

Per una directory invece:

```
sudo unmount "punto di mount"
```

Il comando seguente mostra come ottenere l'elenco dei dispositivi montati:

```
roberto@roberto:~$ sudo mount
[sudo] password for roberto:
/dev/sda1 on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
sysfs on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
udev on /dev type devtmpfs (rw,mode=0755)
devpts on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,noexec,nosuid,nodev,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs (rw,noexec,nosuid,nodev,size=104857600,mode=0755)
```

La file name resolution è il procedimento con cui si individua il contenuto del file a cui il percorso fa riferimento esaminando il pathname da destra a sinistra e localizzando ogni nome nella directory indicata dal nome precedente usando / come separatore.
Se il percorso comincia per / la ricerca parte dalla directory radice del processo che è la stessa per tutti i processi ed equivale alla directory radice dell'albero dei file: in questo caso si parla di un percorso assoluto. Quando invece il percorso fa riferimento alla directory corrente il pathname è detto percorso relativo. Gli identificatori punto (.) e punto punto (..) hanno un significato speciale e vengono inseriti in ogni directory, il primo fa riferimento alla directory corrente e il secondo alla directory genitrice (o parent directory) cioè la directory che contiene il riferimento alla directory corrente.
Tutti i file sono identificati da un nome che deve essere univoco per tutti quelli dello stesso livello, cioè figli dello stesso nodo. È opportuno sottolineare il fatto che il nome di un oggetto del filesytem è case sensitive, il sistema pertanto fa differenza tra le lettere maiuscole e quelle minuscole utilizzate nel nome dei file. Per esempio i nomi esermate e eserMate rappresentano due file diversi.

I tipi di file

I file sono implementati come oggetti del Virtual File System (VFS) e sono classificati come indicato di seguito. Questa classificazione è basata sul VFS e non sul contenuto, o sul tipo di accesso. Una delle principali differenze di Linux con altri sistemi operativi, come per esempio Windows, è che tutti i file di dati sono identici e contengono un flusso continuo di byte. Vediamo come vengono classificati i file secondo lo standard chiamato FSH (Filesystem Hierarchy Standard):

Tipo di file		Descrizione
regular file	*file normale*	Un file che contiene dei dati (l'accezione normale di file)
directory	*cartella o direttorio*	Un file che contiene una lista di nomi associati a degli *inode*
symbolic link	*collegamento simbolico*	Un file che contiene un riferimento a un altro file/directory
char device	*dispositivo a caratteri*	Un file che identifica una periferica ad accesso sequenziale
block device	*dispositivo a blocchi*	Un file che identifica una periferica ad accesso diretto
fifo	*tubo*	Un file speciale che identifica una linea di comunicazione software (unidirezionale)
socket	*manicotto*	Un file speciale che identifica una linea di comunicazione software (bidirezionale)

■ Le directory secondo l'FHS

Come visto in precedenza FHS (Filesystem Hierarchy Standard) è uno standard che fa riferimento al sistema operativo Unix da cui deriva. Non è rigido, infatti può variare da distribuzione a distribuzione, nella seguente immagine vediamo la struttura di directory della distribuzione Ubuntu:



Analizziamo il significato di ciascuna di esse.

/bin

Contiene i comandi fondamentali per il funzionamento basilare del sistema, è utilizzabile sia dall'amministratore che dagli utenti e non contiene sotto directory. I comandi verranno ampiamente discussi nella prossima unità e sono formati da parole chiave come indicato di seguito:

```
ls
mount
fdisk
prted
chmod
cat
...
```

/boot

Contiene i dati necessari per la fase di avvio del sistema, quali per esempio le copie dell'MBR. Il kernel deve essere posizionato in questa directory o direttamente nella directory affinché possa essere montato. Non contiene i file di configurazione che devono invece essere inseriti nella direc-

tory /etc), e neanche i programmi necessari per modificare il boot loader che devono invece essere inseriti nella directory /sbin.

/cdrom
Apparsa nelle ultime versioni di FHS, definisce un punto di momtaggio per i soli cdrom, in alternativa a /dev e /media.

/lib
Contiene tutte le librerie dinamiche, rappresentate da file.so necessarie per l'avvio del sistema e l'esecuzione dei programmi in /bin e /sbin. La sottodirectory modules contiene i moduli del kernel caricabili in fase di esecuzione.

/sbin
Analogamente alla directory /bin contiene comandi utili per l'amministrazione del sistema. Non è utilizzabile direttamente dall'utente, ed è necessaria all'avvio del sistema.
Contiene programmi molto importanti come per esempio: init, swapon, halt, shutdown, fdisk, fsck.*, mkfs.*, ifconfig, route.

/dev
Contiene i file dei dispositivi, i cosiddetti device driver, che mettono a disposizione un'interfaccia standard, che si presenta a tutti gli effetti come un file. Gli esempi di dispositivi sono:
- hda, hdb, hdc, hdd, sda, sdb…
- hda1, hda2…
- ttyS0, ttyS1…
- video0, video1…
- random, rtc.

/etc
Contiene i file che definiscono la configurazione globale della macchina e dei relativi programmi. Contiene alcuni programmi, come per esempio:
- fstab, group, passwd, profile, syslog.conf, ld.so.conf…
- host.conf, hosts, services…
- script di avvio della macchina

Inoltre contiene nella sottodirectory etc/X11 i file di configurazione della scheda grafica.

/home e /root
Home contiene le directory personali degli utenti, secondo la sintassi /home/nomeutente, a eccezione della directory personale dell'amministratore che è posizionata sotto alla directory /root per motivi di sicurezza e affidabilità del sistema. Possiamo fare riferimento alla directory home attraverso il simbolo tilde ("~"). Per esempio l'utente con username roberto avrà come cartella home /home/roberto e per tale utente, la directory ~ rappresenterà appunto la directory /home/roberto.
La home dir viene usata per salvare i file di configurazione a livello utente di vari applicativi.

/lost+found
Contiene gli eventuali file danneggiati trovati dopo un riavvio successivo a un arresto non adeguato del sistema.

/mnt
Viene utilizzata per effettuare il montaggio di file system temporanei, come per esempio le partizioni di altri sistemi operativi, oppure partizioni per memorizzare dati (partizioni di storage).

/media
Viene usata per tutti i dispositivi rimovibili, come per esempio CD e dispositivi di memorizzazione usb.

/opt
È una directory riservata per i pacchetti software aggiuntivi come per esempio pacchetti binari. Contiene i file di applicazioni non fornite dal sistema. È poco utilizzata in Linux, esiste quasi esclusivamente per compatibilità con Unix.

/proc e /sys
La directory /proc da accesso a varie informazione del sistema, come per esempio la cpu, i processi, il sottosistema di rete, come per esempio nelle sottodirectory /proc/cpuinfo, /proc/version. Possiamo agire su alcuni di questi file per modificare dinamicamente lo stato del kernel. Dalla versione del kernel 2.6 molte informazioni sono state spostate nella directory /sys.

/tmp
Utilizzata per contenere i file temporanei eventualmente richiesti dai programmi. Una volta terminato l'uso di un file temporaneo, esso può essere cancellato in qualsiasi momento, in quanto viene cancellata all'avvio.

/usr
Ha una struttura formata da numerose sottodirectory, analogamente alla directory radice (/). Non è modificabile dall'utente e nessun programma può creare delle sottocartelle al suo interno. Contiene alcune sottodirectory interessanti:
- /usr/bin, /usr/sbin, /usr/lib: corrispondono dal punto di vista funzionale alle rispettive cartelle del filesystem radice, e al contrario di queste contengono programmi non vitali per il funzionamento del sistema.
- /usr/games: contiene giochi e programmi educativi
- /usr/include: contiene i file include (.h) usati dai programmi C/CPP
- /usr/local: utilizzato per creare un'ulteriore livello di gerarchia. Per esempio, possono essere posizionati qui i programmi compilati sul sistema, senza utilizzare un packet manager
- /usr/share: contiene informazioni documentali come per esempio i manuali dei comandi, mappe e icone
- /usr/src: contiene i sorgenti dei programmi, anche del kernel

/var
Contiene tutte le informazioni variabili, come per esempio le directory di spool, i file di log e le informazioni sui programmi in esecuzione e gli archivi dei database server. Contiene alcune interessanti sottodirectory:
- /var/cache: usata come per la cache delle applicazioni che possono eventualmente creare sottocartelle per i propri dati
- /var/lib: contiene informazioni relative ai programmi, come per esempio, gli archivi dei motori di database postgresql e mysql che sono presenti nelle sottodirectory: /var/lib/postgres e /var/lib/mysql.
- /var/log: contiene i file di log
- /var/mail: contiene le mailbox degli utenti, se presente un server di posta
- /var/run: contiene file che descrivono lo stato corrente del sistema. Per esempio, per ogni processo in esecuzione, in questa directory si trova un file in formato.pid che contiene l'identificativo del processo
- /var/spool: contiene i dati della coda di processo, per esempio della stampante nella sottodirectory /spool/lpd

La struttura fisica del file system

L'unità di memorizzazione più piccola che è in grado di gestire il file system di Linux è il ◀ blocco ▶.



◀ **Blocco** Un blocco (dall'inglese *block*) è una sequenza di byte che memorizzati sulla memoria di massa. Il file system di Linux accede ai file passando per i blocchi. ▶

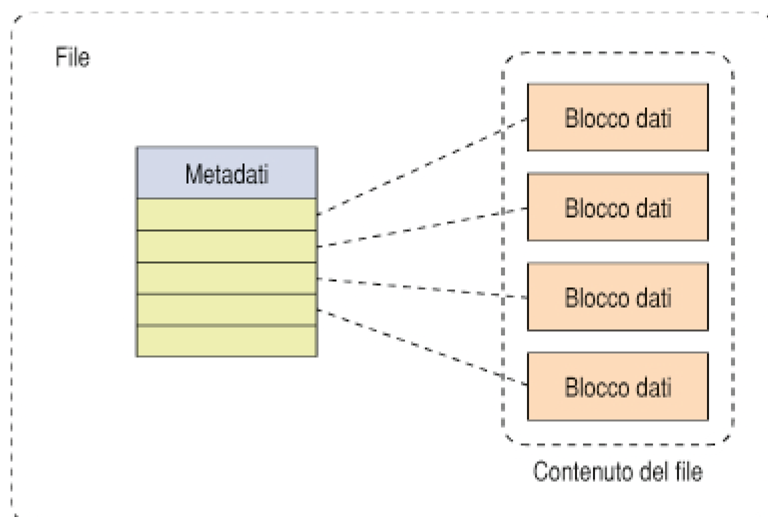
Ciascun blocco del disco può trovarsi in due situazioni:
- allocato (allocated);
- libero (free).

Nel primo caso il blocco contiene effettivamente delle informazioni, mentre nel secondo caso non contiene alcun dato.
La dimensione dei blocchi può essere scelta e modificata in relazione alla partizione: blocchi di dimensione minima riducono gli sprechi di memoria ma aumentano i tempi di accesso alle informazioni, viceversa aumentano gli sprechi di memoria a vantaggio delle prestazioni in lettura/scrittura.

Non si confondano i blocchi logici con quelli fisici: i primi contengono le informazioni del file system, i secondi vengono utilizzati per conservare i dati veri e propri sulla superficie magnetica del dispositivo di memoria di massa. In genere i blocchi fisici vengono identificati dai settori che hanno una dimensione di 512 byte, mentre i blocchi logici di Linux hanno una dimensione maggiore, solitamente un multiplo di quella dei blocchi fisici (1 kB, 2 kB, 4 kB).

Il file system ext

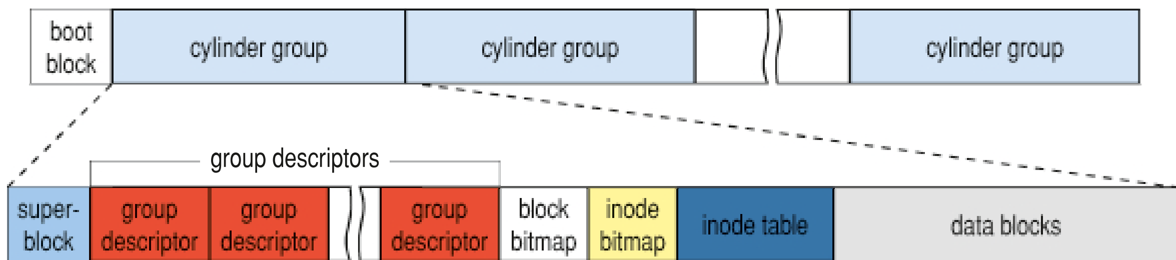
Adesso vediamo la struttura del file system GNU/Linux ext (Extended Filesystem), che esiste in 3 versioni: ext2, ext3 ed ext4. Ciascun file viene rappresentato da un'apposita struttura che ne contiene le caratteristiche e il riferimento ai blocchi che contengono i dati del file stesso, come indicato nella figura seguente:



Come abbiamo già anticipato l'unità minima di memorizzazione dei dati è il blocco logico, che in Linux è di dimensione pari a 1 kB, 2 kB oppure 4 kB e viene decisa in fase di creazione del file system.

I blocchi di Linux sono paragonabili ai cluster di Windows.

Per ciascuna partizione, che il sistema identifica come cylinder group, vi sono diverse informazioni, così riassunte:



Come possiamo notare i blocchi vengono raggruppati all'interno dei cylinder group in modo da limitare gli eventuali errori all'interno del singolo cylinder group e ridurre i tempi di accesso memorizzando i file nelle vicinanze delle directory che li contengono. In ciascun cylinder group vi è un superblock che racchiude le seguenti informazioni:

- caratteristiche tipiche (identificazione del file system, dimensioni, struttura);
- parametri modificabili (massimo numero di mount);
- stato (stato di montaggio del file system, numero di blocchi liberi, numero di mount correnti).

Le caratteristiche descrivono la struttura logica, vengono stabilite al momento della creazione e non possono più essere modificate e indicano la dimensione dei blocchi, il numero di blocchi e il numero di inode.

I parametri modificabili possono essere modificati dal ◀ superuser ▶ attraverso il comando terminale tune2fs, mediante il quale si possono variare per esempio i comportamenti del sistema in caso di errore.



◀ Superuser È una sorta di utente amministratore, paragonabile all'Administrator di Windows, possiede alcune caratteristiche che lo rendono in grado di gestire liberamente il sistema. ▶

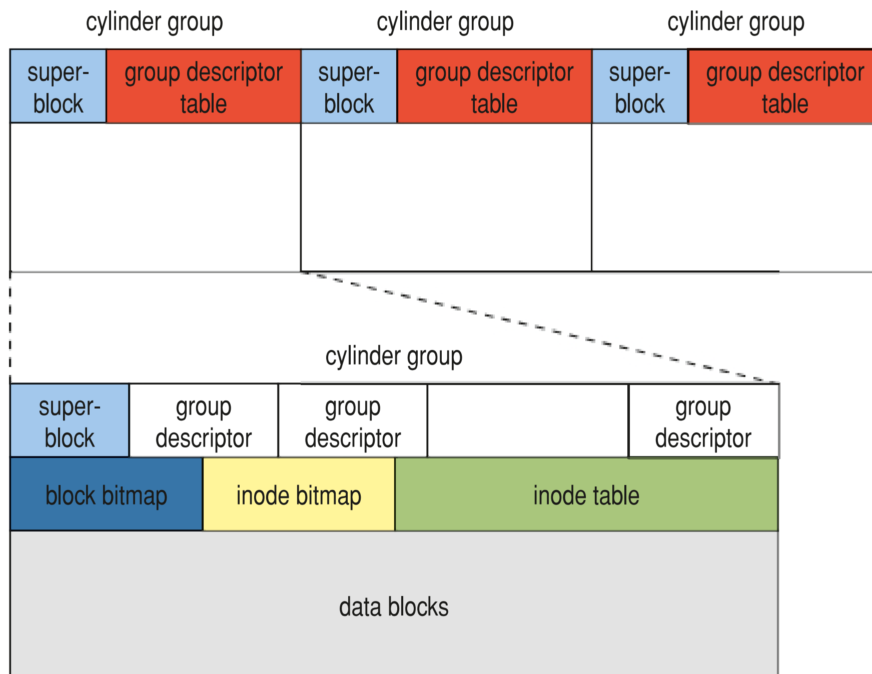
Lo stato del sistema è determinato invece da altre informazioni che vengono aggiornate automaticamente man mano che il file system viene usato, come per esempio il numero di blocchi liberi, lo stato (montato o meno), oppure il numero di volte che esso è stato montato.

Il superblock non possiede alcuna informazione relativa alla partizione che lo ospita, in tal modo il file system risulta totalmente indipendente. Il file system si basa infatti su un modello lineare della partizione sottostante che viene vista come un insieme di blocchi omogenei demandando al device driver tutte le operazioni necessarie per accedere a un singolo blocco logico.

I cylinder group

Come abbiamo visto in precedenza il file system ext è suddiviso logicamente in più parti, chiamate cylinder group, che vengono gestite come entità separate e autonome pur facendo parte della stessa struttura. Lo scopo di questa suddivisione è duplice, innanzi tutto vogliamo minimizzare le conseguenze di eventuali errori, per evitare che si propaghino a tutto il sistema. Il secondo motivo è invece la tendenza a localizzare i files nell'intorno delle loro directory per ridurre i tempi di accesso, cosa che viene ottenuta cercando di allocare inode e blocchi nello stesso cylinder group delle directory. Per ridurre ulteriormente le possibilità che un intero file system venga corrotto a causa

di eventuali errori, sia il superblock che le group descriptor table vengono duplicati in ogni cylinder group, come mostrato in figura:



In questo modo se uno dei superblock o group descriptor viene corrotto a causa di qualche errore esso può essere facilmente ripristinato a partire da una delle sue copie. Abbiamo quindi una struttura altamente ridondante che permette il recupero di eventuali errori.

Come possiamo notare dalla figura precedente ciascun cylinder group contiene una block bitmap che indica quali blocchi del cylinder group sono stati allocati a files o directories, una inode bitmap che indica analogamente quali inode risultano allocati e una inode table che contiene gli inode appartenenti al cylinder group. Ciascuna bitmap è organizzata come una sequenza di bit ognuno dei quali indica se il corrispondente inode o blocco è libero o occupato. Ciascuna bitmap occupa esattamente un blocco e quindi il numero massimo di inodes o di blocchi che possono essere contenuti in un cylinder group è dato dalla dimensione in bytes di un blocco moltiplicata per 8, per esempio con blocchi da 2 kB vi sono 16384 (2048 * 8) blocchi per cylinder group.

Dato che i blocchi e gli inodes sono ripartiti nei vari cylinder group, per individuare la posizione di uno di essi a partire dal suo numero occorre prima individuare il cylinder group, dividendo il numero per il numero di oggetti per cylinder group e quindi individuare l'oggetto all'interno del cylinder group utilizzando come indice il resto di tale divisione. Se per esempio vogliamo accedere all'inode 17654, con 1016 inodes per group si ottiene:

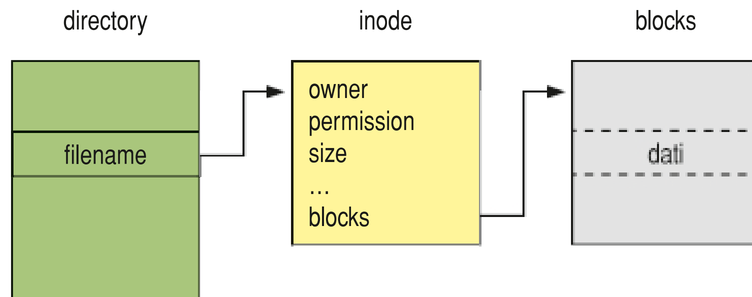
group = 17654 / 1016 = 17
index = 17654 – (17 x 1016) = 382 = resto della divisione

pertanto dovremo cercare l'inode 382 del cylinder group 17. Ciascun gruppo contiene solo una parte degli inodes e dei blocchi di tutto il file system e l'indice ottenuto vale solo all'interno del gruppo.

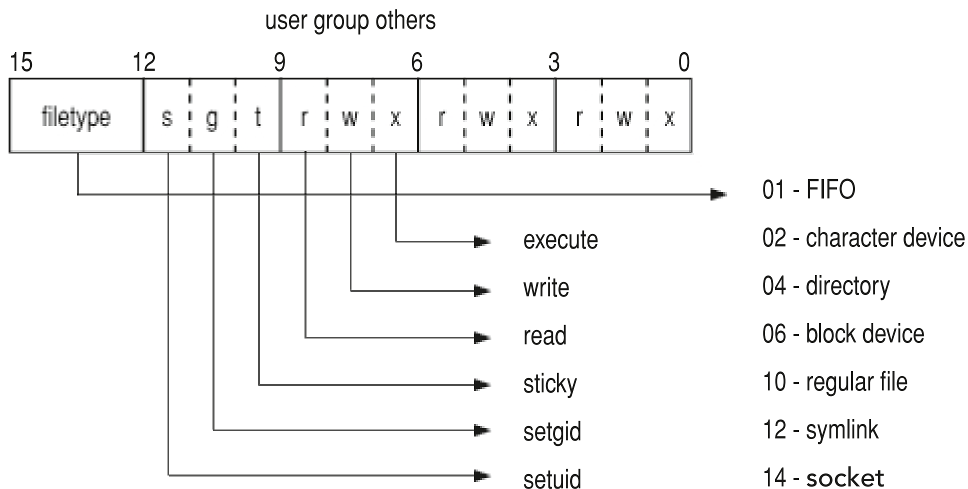
Ciascun cylinder group è descritto da un apposito blocco di controllo chiamato group descriptor che contiene, oltre ai puntatori alle bitmap di allocazione e alla tabella degli inode, anche dei contatori che sono aggiornati e utilizzati dalle routines di allocazione.

Gli inodes

L'inode è la risorsa principale associata a ogni file o directory, che ne identifica le caratteristiche e indica dove sono fisicamente memorizzati i dati. Tutte le operazioni che possono essere effettuate su di un file, vengono in realtà effettuate tramite il suo inode, che contiene tutte le informazioni sul file stesso, esclusi i dati veri e propri, come sintetizzato dalla seguente figura:



Gli inode vengono numerati in maniera progressiva partendo da 1, il numero che li rappresenta viene usato dal kernel per accedere all'inode e quindi ai dati del file corrispondente. Come possiamo notare dalla figura precedente, l'inode contiene molte informazioni che descrivono le caratteristiche di un file, come per esempio i permessi e la dimensione, oltre al puntatore del blocco che contiene fisicamente i dati. Il primo campo dell'inode, chiamato imode contiene le seguenti informazioni:

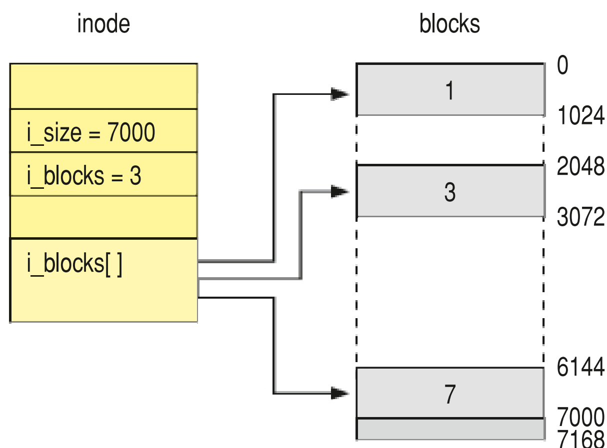


Possiamo notare che i 4 bit più significativi indicano il tipo di file, quindi i permessi suddivisi per gli utenti e i gruppi di utenti. Tra gli altri campi dobbiamo conoscere il significato del campo chiamato iblocks che indica il numero di blocchi occupati dal file, compresi i blocchi di indirizzamento indiretto che verranno illustrati più avanti. Il campo isize contiene la dimensione del file oppure, nel caso di un device, un numero a 16 bit che specifica il tipo di device.

I significati delle varie combinazioni di flag dei permessi sono riportati nella tabella seguente nella quale abbiamo indicato con il simbolo "–" il fatto che il valore del bit relativo è ininfluente rispetto a quanto indicato in ciascuna riga:

			owner user			owner group			others			Significato
s_u	s_g	t	r_u	w_u	x_u	r_g	w_g	x_g	r_0	w_0	x_0	
1	–	–	–	–	–	–	–	–	–	–	–	Se eseguito, il relativo processo ha i permessi del proprietario
–	1	–	–	–	–	–	–	–	–	–	–	Se eseguibile, quando eseguito, il relativo processo ha i permessi del gruppo proprietario Se non eseguibile, indica che il *mandatory locking* è attivato
–	–	1	–	–	–	–	–	–	–	–	–	Non utilizzato
–	–	–	1	–	–	–	–	–	–	–	–	Permesso di lettura per il proprietario
–	–	–	–	1	–	–	–	–	–	–	–	Permesso di scrittura per il proprietario
–	–	–	–	–	1	–	–	–	–	–	–	Permesso di esecuzione per il proprietario
–	–	–	–	–	–	1	–	–	–	–	–	Permesso di lettura per il gruppo proprietario
–	–	–	–	–	–	–	1	–	–	–	–	Permesso di scrittura per il gruppo proprietario
–	–	–	–	–	–	–	–	1	–	–	–	Permesso di esecuzione per il gruppo proprietario
–	–	–	–	–	–	–	–	–	1	–	–	Permesso di lettura per tutti gli altri utenti
–	–	–	–	–	–	–	–	–	–	1	–	Permesso di scrittura per tutti gli altri utenti
–	–	–	–	–	–	–	–	–	–	–	1	Permesso di esecuzione per tutti gli altri utenti

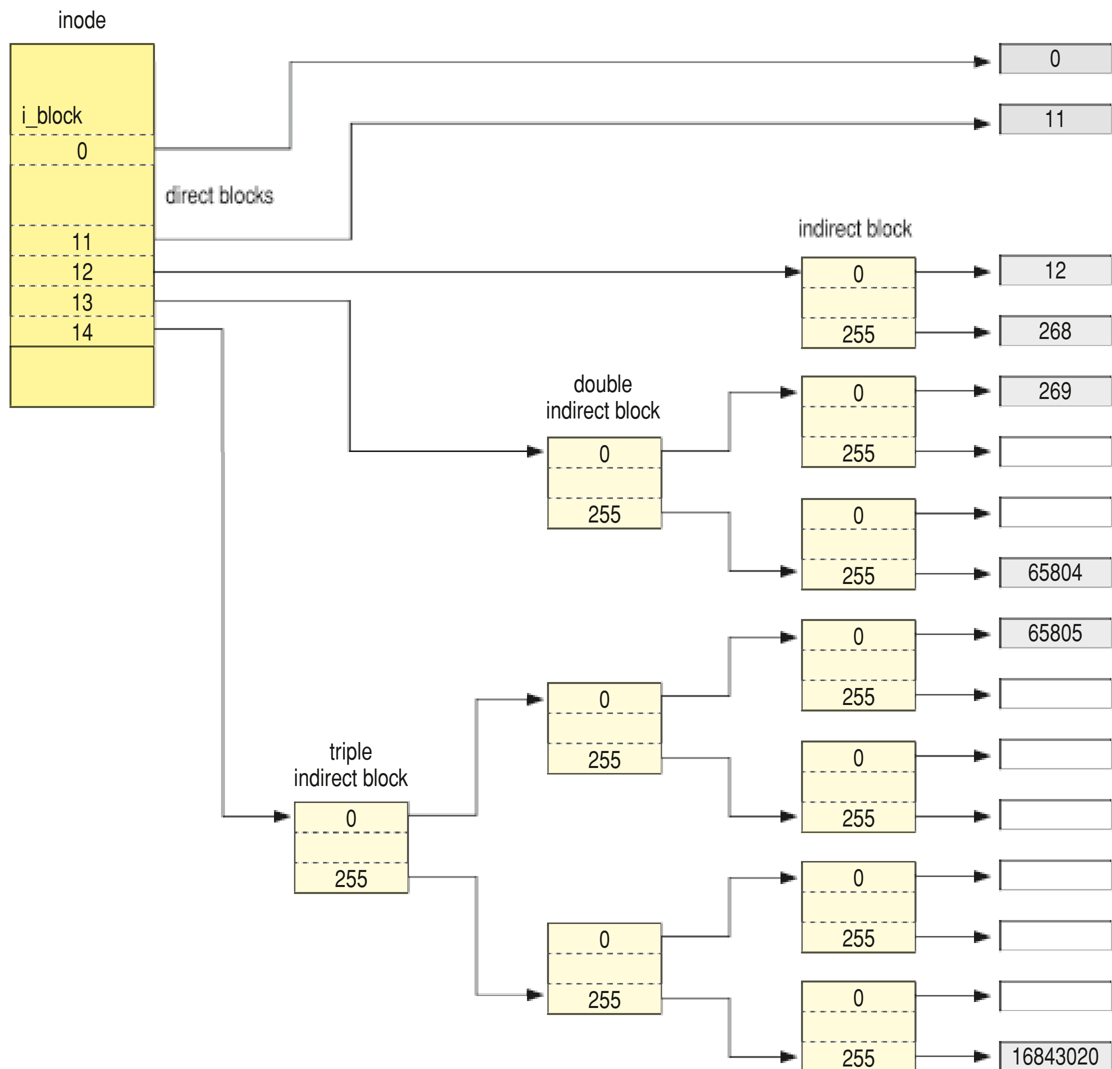
I campi isize e iblocks possono indicare dimensioni diverse dello stesso file, questo può accadere in quanto possono essere creati degli sparse files, cioè files che non possiedono tutti i blocchi allocati.
Nella figura seguente è illustrato un file sparso di 7000 byte che occupa tuttavia 3 soli blocchi da 1024 bytes. Solo i blocchi di dati corrispondenti agli offset 0, 2048 e 6144, sono allocati, gli altri no: ▶



Il campo iblocks è in realtà un array di 15 elementi che puntano ai blocchi di dati del file. I primi 12 elementi, da iblocks[0] a iblocks[11], vengono chiamati direct blocks e contengono i numeri dei primi 12 blocchi del file. L'elemento successivo, iblocks[12], punta a un blocco chiamato indirect block, che non contiene dati ma un ulteriore array di puntatori ai blocchi di dati.
Sapendo che ciascun blocco ha una dimensione pari a 1024 Byte, potrà contenere 256 (1024/4) diversi puntatori a blocchi di dati, in modo da poter indirizzare i blocchi da 12 a 268.
L'elemento iblocks[13] punta a un altro blocco chiamato double indirect block, che a sua volta punta a un blocco di puntatori a blocchi di puntatori, in grado quindi di indirizzare 256*256 blocchi di dati, quindi da 269 a 65804. L'ultimo elemento, iblocks[14] viene chiamato triple indirect block, che aggiunge un ulteriore livello di indirizzamento consentendo l'indirizzamento di altri 256*256*256 blocchi di dati oltre il blocco 65804.

Ovviamente tutti questi blocchi indiretti vengono allocati solo se sono effettivamente necessari per indirizzare dei blocchi di dati.

La rappresentazione dello schema di puntatori dell'inode è il seguente:



Il vantaggio fornito dall'organizzazione di Linux è quello di far risparmiare spazio allocato per file di dimensione ridotta, dato che per i primi 12 kB sono sufficienti i puntatori diretti contenuti nell'i-node. In tal modo si risparmia anche l'accesso al disco che sarebbe necessario per leggere il blocco, man mano che il file cresce di dimensioni è invece necessario introdurre livelli successivi di indirizzamento indiretto per contenere i puntatori ai dati.

Se confrontiamo questa organizzazione con quella di Windows possiamo notare che distribuendo i puntatori ai dati nelle vicinanze dei dati stessi si migliorano i tempi di accesso, mentre in Windows è sempre necessario muovere le testine del disco all'inizio della partizione per poter leggere i puntatori ai cluster contenuti nella tabella di allocazione dei file.

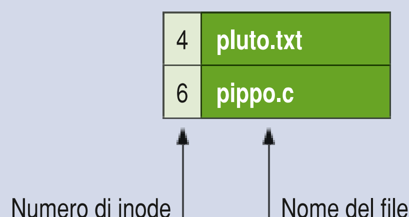
L'allocazione dello spazio per i dati viene effettuata mediante un algoritmo che cerca di utilizzare blocchi contigui minimizzando così la frammentazione dei dati che è invece tipica dei sistemi operativi più primitivi. Linux utilizza dei contatori che indicano in ogni istante lo spazio libero per ciascun cylinder group e delle bitmap di allocazione che permettono di cercare velocemente dove conviene allocare nuovo spazio per i dati. Contatori analoghi vengono mantenuti per inode e directory in modo da poter distribuire i dati uniformemente per tutto il file system.

La directory

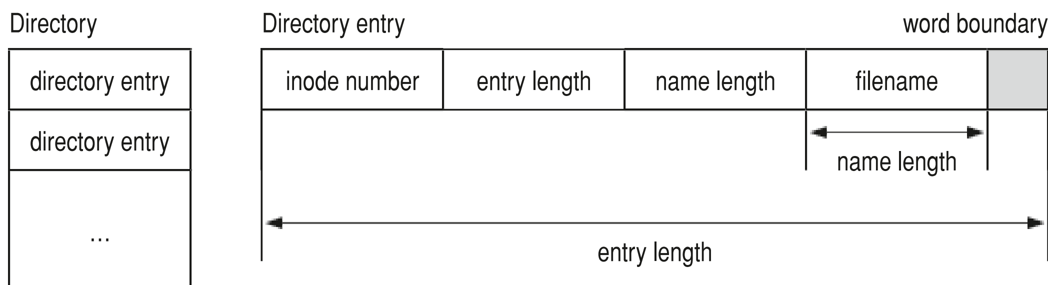
Una directory è, come abbiamo già sottolineato, un file che possiede una caratteristica: i dati in esso contenuti sono le informazioni sui file nella directory. Ciascuna ◀ directory entry ▶ è costituita da un record di lunghezza variabile, contente il nome del file e il numero di inode corrispondente.



◀ Directory entry Una directory entry, contiene il nome del file associato al corrispondente inode, come indicato nello schema seguente:

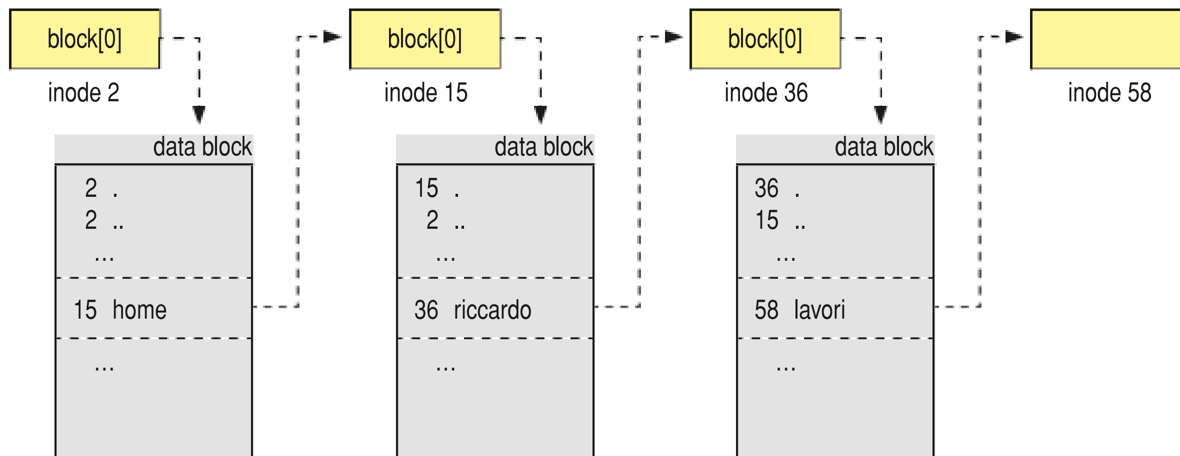


Le altre informazioni relative al file sono indicate dall'inode, la directory serve solo per collegare il nome del file col suo inode. La struttura di una directory entry tipica è indicata dalla figura seguente:



L'identificazione di un file da parte del kernel incomincia sempre dal directory entry. Quando specifichiamo il percorso di un file prima di tutto viene effettuata una ricerca del nome del file nelle directory entry delle varie directory che costituiscono il path del file. Man mano che il nome del

file viene trovato, vengono recuperati i dati presenti nel blocco (o nei blocchi) puntato dall'inode associato. La figura seguente mostra un esempio di accesso a un file:



Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Linux è:

a) un sistema operativo
b) un kernel
c) un bootloader
d) una partizione

2 I file di dispositivo sono contenuti nella directory:

a) /dev b) /home c) /bin d) /etc

3 Quale comando tra i seguenti consente di modificare le partizioni esistenti:

a) mount b) gparted c) sudo d) fdisk

4 Le partizioni non possono essere:

a) estese b) logiche c) primarie d) secondarie

5 Quale directory tra le seguenti contiene i comandi fondamentali per il funzionamento basilare del sistema:

a) /dev
b) /boot
c) /lib
d) /bin
e) /etc

6 Quale directory tra le seguenti contiene i file che definiscono la configurazione globale della macchina:

a) /bin b) /lib c) /etc d) /config

7 Quale directory tra le seguenti non è modificabile dall'utente e nessun programma può creare delle sottocartelle al suo interno?

a) /usr b) /var c) /sbin d) /mnt

Test vero/falso

1 Il partizionamento è nato dal fatto che i sistemi operativi più vecchi non riuscivano a gestire dischi grandi. V F
2 Il S.O. Linux deve avere a disposizione minimo due partizioni. V F
3 Lo swap è una partizione di dimensione almeno doppia rispetto alla partizione tradizionale. V F
4 La file name resolution è il procedimento con cui si individua il contenuto del file a cui il percorso fa riferimento esaminando il pathname da destra a sinistra. V F
5 Linux possiede un file system non case sensitive. V F
6 L'unità di memorizzazione più piccola gestita da Linux è il blocco. V F
7 Il file system di Linux accede ai file passando per i blocchi. V F
8 In ogni cylinder group vi è un superblock che racchiude le informazioni sullo stato del montaggio del file system. V F
9 L'inode contiene i dati del file. V F
10 L' iblocks indica il numero di blocchi occupati dal file. V F
11 Le directory entry collegano solo il nome del file al suo inode. V F

Esercizi di completamento

1 Le distribuzioni si compongono di due elementi di base: il .... che gestisce il funzionamento dei componenti del sistema e .... che gestisce l'interfaccia grafica del sistema operativo

2 I tre componenti che caratterizzano le distribuzioni sono: l'...................., la .... e le ....

3 I pacchetti vengono gestiti dal .... che ha il compito di automatizzare il processo di ...., aggiornamento, ... e rimozione dei pacchetti software.

4 In Linux tutti i file sono conservati in un albero la cui radice si chiama .... viene montata all'...................., inoltre ogni file è identificato dal proprio .... composto da una serie di nomi separati dallo ..../.............

5 Quando un percorso inizia con il simbolo "/" si parla di un percorso .... Quando invece il percorso fa riferimento alla directory corrente il percorso è detto ....

6 I file sono implementati come oggetti del ....

7 I blocchi logici contengono le ...., i blocchi fisici vengono utilizzati per .... nella ....

8 I blocchi fisici vengono identificati dai .... che hanno una dimensione di ...., mentre i blocchi logici di Linux hanno una dimensione pari a un ....

9 L'inode identifica le caratteristiche dei file e indica dove sono .... Tutte le operazioni che possono essere effettuate su di un file, vengono in realtà effettuate tramite il suo .... che contiene tutte le informazioni sul file stesso, esclusi i ....

10 Il campo .... contiene la dimensione del file oppure, nel caso di un device, un numero a 16 bit che specifica il tipo di device

11 Una directory entry è costituita da un record di lunghezza ...., contente il nome del .... e il numero del suo ....

Esercitazioni di Laboratorio 1

L'INSTALLAZIONE DI UBUNTU

Scaricare Ubuntu

Ubuntu è una distibuzione di Linux assai diffusa e possiede una interfaccia grafica GUI in stile Windows mediante la quale possiamo svolgere quasi tutti tutte le principali funzioni eseguibili mediante l'ambiente di ◀ shell ▶.



◀ Shell La shell di un ambiente Linux è un interprete a linea di comando che permette di interagire con il sistema operativo, impartendogli dei comandi di sistema fino a creare un programma vero e proprio, generalmente chiamato script. ▶

Alla prima installazione Ubuntu si presenta con un tema grafico assai semplice ed essenziale, personalizzabile dall'utente. Prima di tutto però dobbiamo procurarci una versione del file da installare, la seguente procedura illustra come scaricare Linux nella distribuzione Ubuntu:

1. Per prima cosa dobbiamo connetterci al sito: http://www.ubuntu-it.org/download, all'interno di questo sito dobbiamo selezionare il tipo di distribuzione che intendiamo scaricare, in questo caso si tratta di una versione desktop, l'altra disponibile è la versione server. Quindi dobbiamo selezionare la versione (in questo caso la 12.10), quindi il tipo di piattaforma hardware (32 o 64 bit).

2. Una volta effettuata la scelta facciamo click su Avvia download: ▶



3. Il file che viene ottenuto è di tipo ISO: ▼



Il file che abbiamo scaricato può essere usato per creare un CD avviabile, oppure una pendrive (chiavetta) avviabile e in ultimo può essere anche usato per provare e/o installare Ubuntu in Macchina virtuale.

Prima di installare Linux dobbiamo verificare l'esistenza di una partizione libera di almeno 10 gigabyte, la partizione di swap verrà creata automaticamente da Ubuntu durante la fase di installazione.

Installare Linux da CD

Per creare un CD dal quale far partire l'installazione di Ubuntu è assai semplice, possiamo farlo semplicemente masterizzando il file immagine che abbiamo scaricato in precedenza (.iso) su di un CD mediante un software di masterizzazione come per esempio Nero Burning. Successivamente dovremo inserire il CD e quindi riavviare il sistema.

Installare Ubuntu da pendrive

In molti casi può essere utile poter installare Ubuntu da una qualunque chiavetta USB, al posto del CD. Per installare Ubuntu da pendrive, dobbiamo innanzi tutto formattare la chiavetta, quindi utilizzare un software in grado di rendere avviabile il supporto USB, come per esempio ◀ UNetbootin ▶.

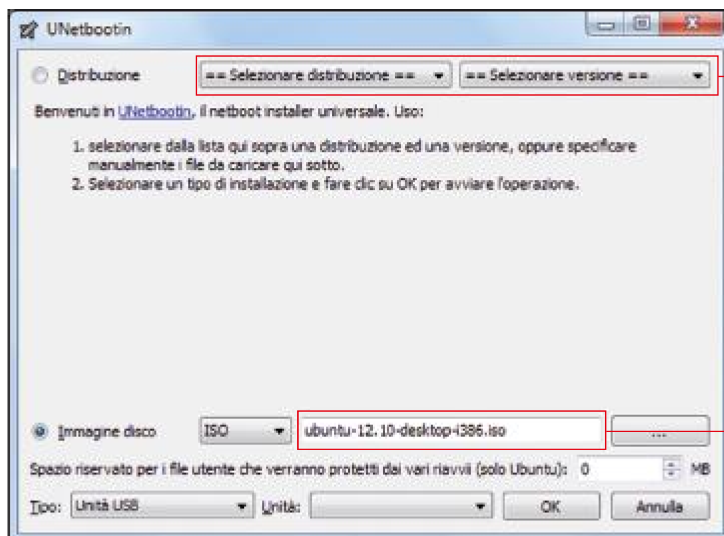


◀ UNetbootin UNetbootin è un software gratuito che consente di rendere avviabile una pendrive. Tale software esiste nelle varianti per Windows, Linux e Mac. Supporta moltissime distribuzioni Linux e permette di trasferire, su chiavetta avviabile, una ISO di un CD-DVD avviabile qualsiasi. ▶

Per procurarci il software UNetbootin, dobbiamo connetterci al sito: http://unetbootin.sourceforge.net/: ▶



Il file che scarichiamo è direttamente eseguibile. Dobbiamo infatti fare doppio click su di esso e connettere la chiavetta USB dopo averla formattata FAT32. La finestra principale del programma è la seguente:



Selezionare la distribuzione e la versione per installare direttamente Linux su chiavetta (versione Live)

Selezionare il file .iso della distribuzione da copiare sulla chiavetta

La parte superiore della finestra consente di selezionare la versione di Linux da installare sulla chiavetta per effettuare un utilizzo in ◀ modalità Live ▶.
Nella parte inferiore della finestra possiamo selezionare il file.iso da copiare nella chiavetta USB, in modo da lanciare successivamente l'installazione, riavviando il sistema e inserendo la pendrive dopo aver modificato la sequenza di boot da BIOS. In questo caso abbiamo selezionato la distribuzione di Ubuntu che abbiamo scaricato in precedenza.



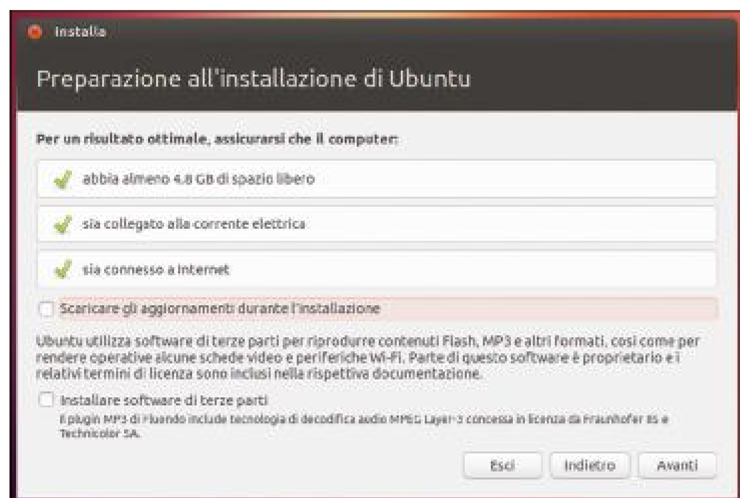
◀ Modalità Live Si tratta della modalità più semplice per provare a utilizzare Ubuntu senza che venga installato fisicamente sul disco fisso. In questa modalità il sistema operativo rimane sul supporto estraibile (pendrive o CD) per essere caricato direttamente dal supporto in modalità dal vivo, chiamata appunto Live. ▶

La seguente procedura mostra come installare Ubuntu.

1. Per prima cosa dobbiamo riavviare il sistema avendo cura di inserire il supporto (CD o pendrive). La finestra che appare è la seguente: ▶

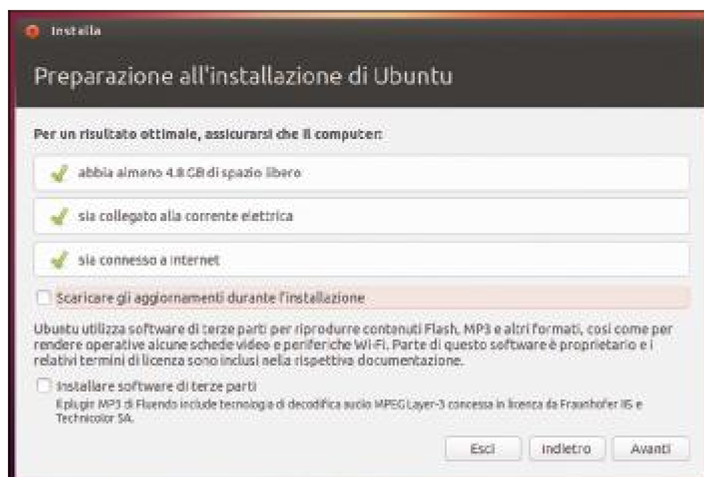


2. Scegliamo Installa Ubuntu per procedere all'installazione completa. ▶

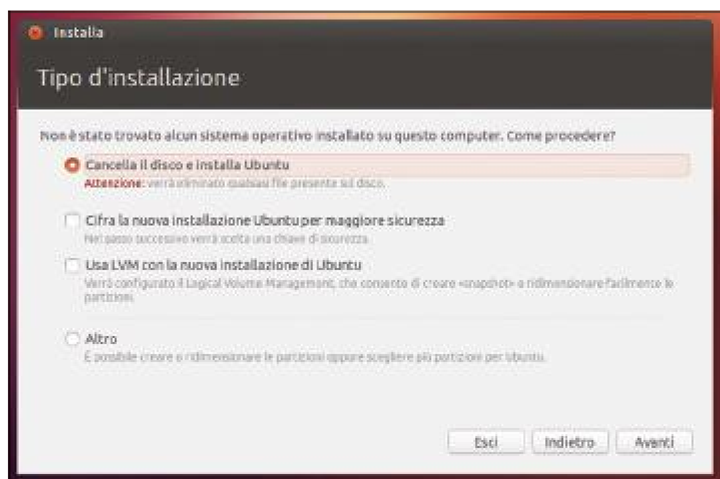


Selezionando la voce Prova Ubuntu effettuiamo una esecuzione di Ubuntu in modalità Live.

❸ A questo punto dobbiamo verificare di avere almeno 4.8 GB di spazio libero sulla partizione che abbiamo individuato per l'installazione di Ubuntu: ▶



❹ Dopo aver fatto click su Avanti otteniamo la seguente finestra: ▶

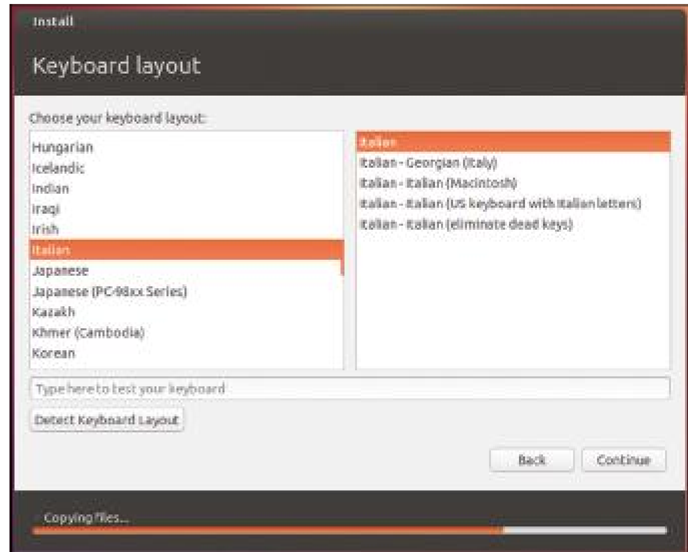


❺ In questo caso selezioniamo Cancella il disco e installa Ubuntu per installarlo nella partizione individuata cancellando l'eventuale contenuto precedente. Quindi fai click su Avanti.

❻ La schermata successiva consente di selezionare la data e l'ora attraverso la scelta del fuso orario: ▶



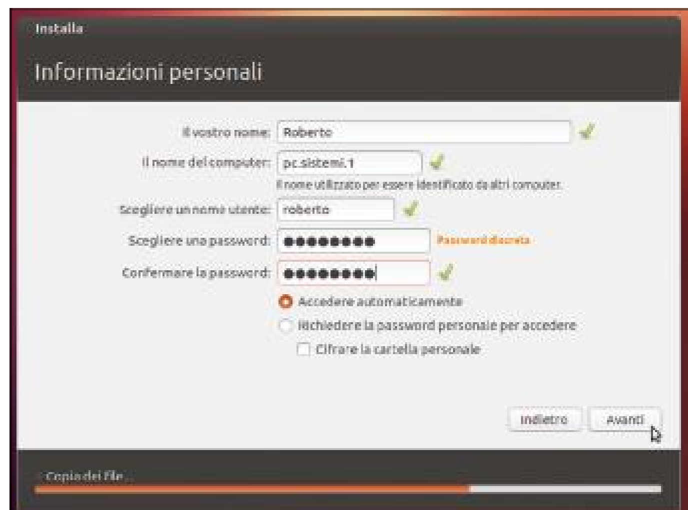
7 Dopo aver fatto click su Continue otteniamo la seguente schermata: ▶



8 Dopo aver scelto la lingua passiamo alla finestra successiva che ci obbliga all'immissione dei dati di login come Amministratore: ▶



9 Nel nostro caso inseriamo i seguenti dati: ▶



Dobbiamo selezionare Richiedere la password personale per accedere quando ci sono più utenti che intendono accedere al sistema.

❿ A questo punto inizia l'installazione vera e propria con la copia dei file di sistema:

```
Welcome to Ubuntu 12.10 (GNU/Linux 3.5.0-17-generic i686)

 * Documentation:  https://help.ubuntu.com/

ubuntu@ubuntu:~$
Broadcast message from root@ubuntu
        (unknown) at 17:32 ...

The system is going down for reboot NOW!
modem-manager[1373]: <info>  Caught signal 15, shutting down...
```

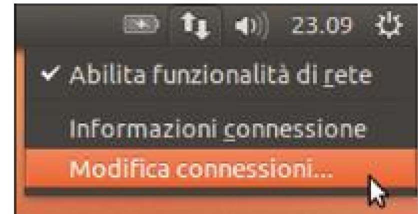
⓫ Al termine appare la finestra principale di Ubuntu:

⓬ A questo punto l'installazione è terminata, proviamo a effettuare la ricerca per utilizzare l'applicazione Terminale. È sufficiente scrivere il nome dell'applicazione nella casella di testo e il sistema provvederà a mostrare l'applicazione trovata:

Modifichiamo le impostazioni della scheda Ethernet

Per modificare le impostazioni della scheda Ethernet di cui è dotato il sistema procediamo nel modo seguente.

1. Facciamo click sul pulsante che mostra due frecce opposte posizionato in alto a destra: ▶

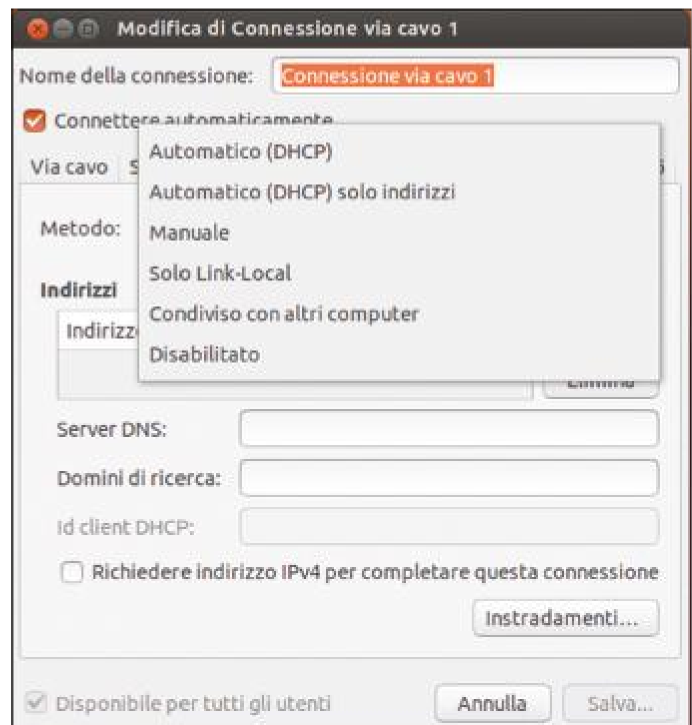


2. A questo punto appare una finestra che mostra le connessioni eventualmente disponibili. Possiamo modificare le connessioni esistenti, crearne una nuova oppure eliminarle: ▶



3. Per creare una nuova connessione facciamo click su Aggiungi..., appare la seguente finestra in cui dobbiamo posizionarci sulla scheda chiamata Impostazioni IPv4: ▶

   Come possiamo notare possiamo innanzi tutto decidere se l'indirizzo IP della connessione deve essere di tipo Dinamico (DHCP) oppure statico (Manuale). Scegliamo il tipo statico e procediamo.

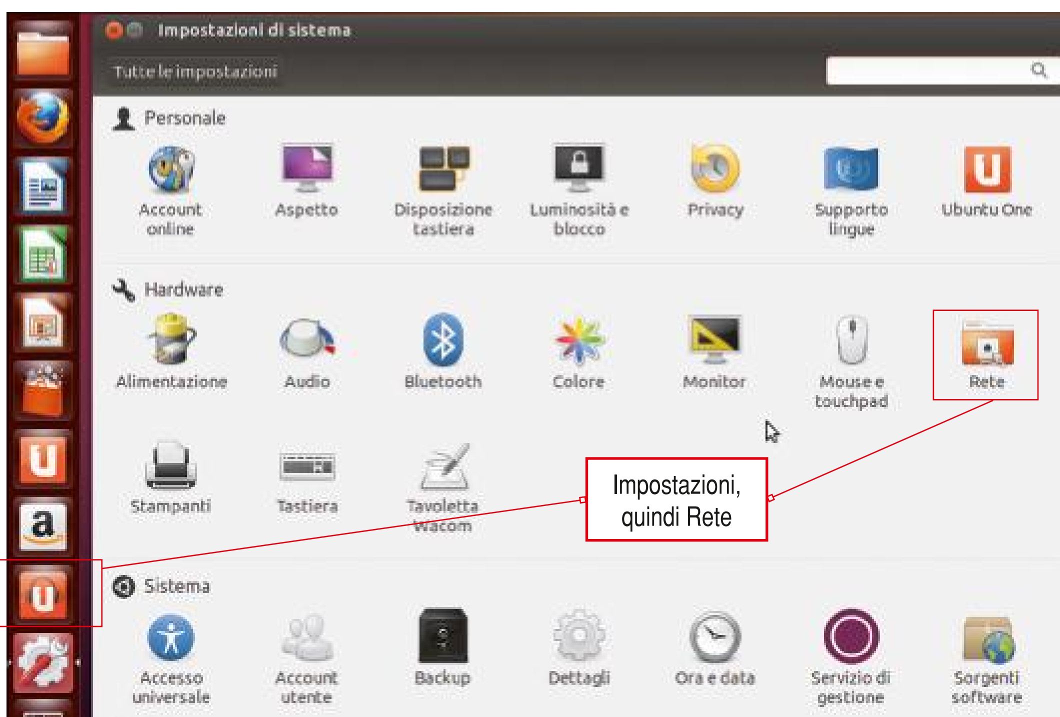


4 Adesso dobbiamo indicare l'indirizzo IP della scheda, della maschera, di un eventuale Gateway e di un Server DNS: ▶

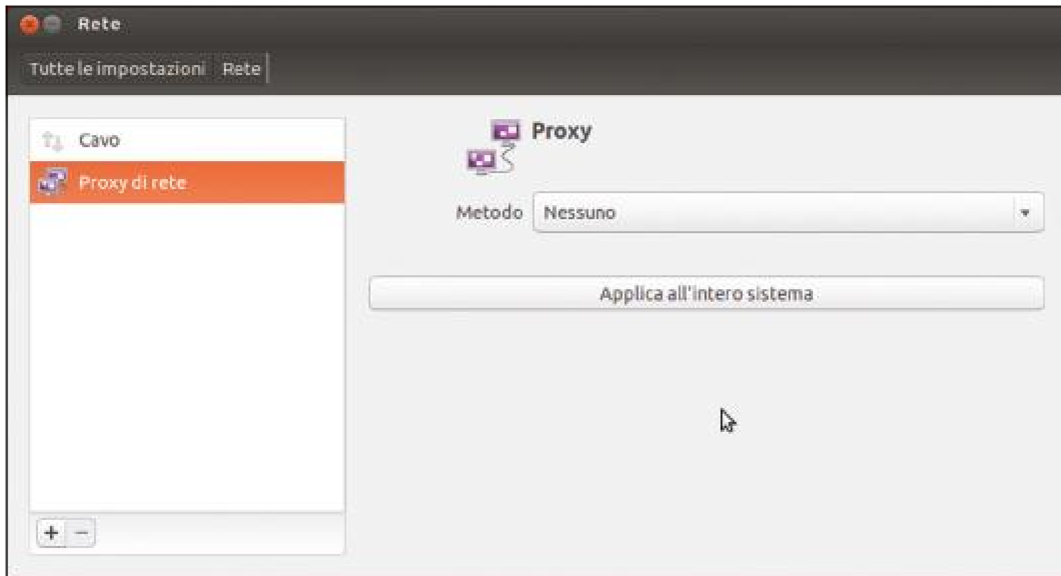


Il settaggio del proxy

Nel caso in cui il sistema fosse dotato di un server proxy, dobbiamo indicarlo agendo sull'icona Impostazioni posta in basso a sinistra, quindi facendo click sull'icona Rete:



In questa finestra si trova la scheda Proxy che consente di indicare l'indirizzo di un proxy server:





Zoom su...

AGGIUNGERE PACCHETTI SOFTWARE

Quando dobbiamo installare pacchetti aggiuntivi e non possiamo usufruire di uno degli strumenti GUI presenti solitamente nelle distribuzioni Ubuntu, come Ubuntu Software Center o synaptic, utilizzabili solo se abbiamo un collegamento veloce a Internet, possiamo procurarci i pacchetti software da installare su un CD o una pendrive, quindi operare come segue:

1. Apriamo la console a riga di comando (Terminale) presente nella sezione Home:



2. Inseriamo il CD oppure la pendrive che contiene il pacchetto da installare

3. Il dispositivo removibile dovrebbe essere montato automaticamente per esempio in /media/cdrom o in /media/usb.

4. Nel caso in cui il dispositivo rimovibile non sia installato, dobbiamo eseguire il comando:

```
$ sudo mount /dev/hdc /mnt
```

oppure:

```
$ sudo mount /dev/sda1 /mnt
```

Il nome del dispositivo da montare da indicare dopo /dev/xxxx, varia in base al dispositivo e alla configurazione del sistema. Per conoscere quale è il nome da usare occorre eseguire il comando:

```
$ dmesg
```

subito dopo aver collegato il dispositivo, osservando le ultime righe di risposta del comando stesso.

5. Eseguiamo quindi il comando seguente:

```
$ cd /mnt
```

e quindi il comando successivo per posizionarci nella directory in cui sono presenti i pacchetti:

```
$ cd dir_dove_sono_i_pacchetti
```

6. Il comando dpkg installerà il pacchetto, come indicato dal seguente comando:

```
$ sudo dpkg --force-all -i *.deb
```

7. Infine dobbiamo rimuovere la penna o il CD, con il seguente comando:

```
$ cd /
$ sudo umount /mnt
```

Esercitazioni di Laboratorio 2

LA SHELL DI UBUNTU

La shell di Linux

Prima di poter iniziare a lavorare con Linux dobbiamo essere consapevoli che esiste una sorta di guscio o per meglio dire ◀ shell ▶, che consente all'utente di interagire con il sistema operativo liberandolo dalla conoscenza approfondita della macchina.



◀ Shell La shell è uno strumento che ci consente di dialogare con la macchina, esiste anche in altri sistemi operativi, con una certa dose di forzatura potremmo dire che anche il Prompt dei comandi di Windows è una shell. La shell è l'interprete dei comandi che legge l'input, lo elabora e restituisce l'output, in termini di esecuzione dei comandi. ▶

La shell è vista da parte dell'utente come un insieme di comandi impartibili al sistema. Ciascun comando possiede un preciso significato ed è rappresentato da una parola chiave che lo identifica, per esempio rm (remove), cd (change directory) ecc. La shell di Linux è rigorosamente case sensitive, cioè distingue caratteri maiuscoli da quelli minuscoli, questo significa per esempio che il file Prova è diverso dal file prova.

Anche se Ubuntu mette a disposizione dell'utente un ambiente GUI, rappresentato dall'uso del mouse, delle finestre e dei menu, non tutti i comandi possono essere eseguiti in modalità grafica, per un controllo completo del sistema è necessario conoscere approfonditamente i comandi impartibili dall'ambiente di shell.

La shell dei comandi è quindi un software che gestisce la comunicazione tra utente e S.O. interpretando ed eseguendo i comandi dell'utente e può essere utilizzata secondo tre diversi livelli di approccio:

- Uso interattivo: il sistema mostra una schermata con sfondo generalmente scuro, a linea di comando (CUI), in cui un prompt mostra che il sistema è pronto a ricevere i comandi dell'utente che devono terminare con la pressione del tasto Enter (Invio).
- Configurazione della sessione: possiamo definire variabili e parametri che vengono utilizzati nell'interazione tra uomo e macchina.
- Programmazione: mediante i comandi di sistema possiamo creare delle routine, chiamate script, attraverso le quali programmare il sistema in modo da automatizzare le operazioni e reagire a eventi.

I comandi non sono altro che programmi contenuti all'interno di alcune directory speciali come le directory /bin e /sbin. Quando l'utente digita comando e preme Invio, la shell cerca tale file nelle directory predefinite, se lo trova lo esegue, altrimenti produce un errore del tipo:

```
bash: nome-comando : command not found
```

Linux mette a diposizione diverse shell, tra cui citiamo:

Shell	Caratteristiche
BOURNE SH	La prima shell di Linux
CSH (C Shell)	Usa una sintassi del tutto simile al linguaggio C
KSH (Korn Shell)	Si tratta del linguaggio di riferimento per tutte le shell, proprietà dell AT&T fino all'anno 2000. Attualmente è una shell di tipo open source e GNU
BASH (Bourne Again Shell)	È la shell standard più completa, inoltre è inserita in quasi tutte le distribuzioni di Linux

> Anche nel vecchio e glorioso MSDOS esiste una shell, si tratta del file COMMAND.COM, che racchiude tutti i comandi interni della relativa shell.

L'interazione con la shell

Linux è un sistema operativo multiutente, significa che più utenti potranno accedere alle stesse risorse e più in generale allo stesso computer. Proprio per questo motivo il sistema è dotato di un sistema di protezione con lo scopo di evitare che altri utenti possano accedere ai nostri file personali.

> Infatti all'avvio del sistema dobbiamo sempre loggarci o più correttamente effettuare una autenticazione inserendo username e password.

Ciascun utente ha accesso alla propria home directory chiamata con la username e rappresentata dal simbolo tilde (~), come possiamo vedere dalla figura seguente: ▶



Come abbiamo visto in precedenza tutte le home directory si trovano nella directory /home e anche tutti i file appartenenti a un particolare utente sono contrassegnati dalla username e potranno essere modificati solo dal proprietario, a meno che non si modifichino i diritti di accesso ai file. Se per esempio proviamo a modificare un file di un altro utente ci rendiamo conto che è effettivamente impossibile.

ESEMPIO 4 *Conoscere il proprietario di un file*

Per conoscere il proprietario di un file possiamo utilizzare il comando ls -l o il comando v che visualizzano il nome dei file contenuti nella directory corrente e l'utente a cui questi appartengono, oltre ad altre informazioni:

```
root@roberto:/home/roberto# ls -l
totale 52
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Musica
-rw-r--r-- 1 root    root      14 gen  7 12:16 prova.txt
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Pubblici
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scaricati
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scrivania
drwxrwxr-x 2 roberto roberto 4096 dic 13 17:46 Ubuntu One
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Video
```

Possiamo notare che esiste, oltre a roberto, un altro utente chiamato root, tra poco vedremo di chi si tratta.
Ciascun utente Linux appartiene a un gruppo a cui è stato assegnato dall'amministratore di sistema durante l'assegnazione dell'account. Per conoscere a quale gruppo apparteniamo possiamo usare il comando id: ▶
Come possiamo notare l'utente appartiene al gruppo root.

```
root@roberto:/home/roberto# id
uid=0(root) gid=0(root) gruppi=0(root)
root@roberto:/home/roberto#
```

I gruppi indicano il tipo di lavoro che abitualmente svolgiamo mediante il sistema, a ciascun gruppo corrispondono determinati diritti di accesso e permessi per utilizzare un insieme di file di sistema.

> Dobbiamo ricordare di chiudere sempre la nostra sessione di lavoro mediante il comando exit, che costringe un utente successivo a reinserire le credenziali.

Per conoscere più approfonditamente la sintassi dei comandi Linux utilizzabili dal Terminale possiamo utilizzare man seguito dal nome del comando, per esempio:

```
man gparted
```

che fornisce spiegazioni dettagliate sul comando gparted, il risultato è il seguente:

```
roberto@roberto: ~
GPARTED(8)                    GParted Manual                    GPARTED(8)

NAME
       gparted - Gnome partition editor for manipulating disk partitions.

SYNOPSIS
       gparted [device]...

DESCRIPTION
       The  gparted  application  is  the GNOME partition editor for creating,
       reorganizing, and deleting disk partitions.

       A disk device can be subdivided  into  one  or  more  partitions.   The
       gparted application enables you to change the partition organization on
       a disk device while preserving the contents of the partition.

       With gparted you can accomplish the following tasks:
       - Create a partition table on a disk device.
       - Enable and disable partition flags such as boot and hidden.
       - Perform actions with partitions such as create, delete, resize, move,
       check, label, copy, and paste.

       More  documentation  can  be  found in the application help manual, and
       online at:
       http://gparted.org

EXAMPLES
       You can run gparted from a command line and specify one  or  more  disk
       devices.

       For  example,  to  start gparted with the devices /dev/sda and /dev/sdc
       you would use the following command:

       gparted /dev/sda /dev/sdc

NOTES
 Manual page gparted(8) line 1 (press h for help or q to quit)
```

Gli hard link ai file

Secondo la rappresentazione fisica del file system di Linux, vista nelle lezioni precedenti, a ciascun file viene associato un record chiamato inode che rappresenta il file e che contiene le seguenti informazioni:

- permessi di accesso;
- utente e gruppo proprietario;
- dimensione;
- data di creazione, dell'ultimo accesso e dell'ultima modifica del file;
- i riferimenti ai settori del disco che contengono i dati fisicamente memorizzati.

Ciascun inode è identificato da un numero univoco nel file system in cui esso risiede. Il numero inode è utilizzato per ottenere l'inode dall'elenco del file system.
La directory è un file speciale che può essere vista come una tabella con due colonne, dove la prima colonna contiene i nomi dei file e la seconda i numeri inode: a ciascuno di questi è associato il nome del file. In sostanza la directory, crea una relazione tra l'inode e il nome del file. Possiamo associare più nomi diversi a un unico file fisico, contrassegnato dal suo numero inode: si parla in questo caso di collegamento fisico al file (◀ hard link ▶).



◀ Hard link A differenza dei soft link in cui abbiamo un file vero e proprio e uno o più link a quel file, gli hard link tra due file puntano allo stesso inode, in tal modo i due files occuperanno su disco lo stesso spazio di un file. Se modifichiamo uno dei due file in realtà li modificheremo entrambi, la rimozione di uno dei file invece non comporterà la cancellazione dei dati dal disco fintanto che altri hard link punteranno allo stesso inode. Quando l'ultimo hard link verrà cancellato allora verranno cancellati anche i dati dal disco. ▶

Un file che possiede 3 hard link significa che è fisicamente collegato a 3 nomi diversi, ma su disco il file è 1 solo. Mediante il comando stat possiamo conoscere le informazioni sull'inode di un file, in questo caso del file prova.txt:

```
roberto@roberto: ~
roberto@roberto:~$ stat prova.txt
  File: "prova.txt"
  Dim.: 14              Blocchi: 8          Blocco di IO: 4096   file regolare
Device: 801h/2049d      Inode: 669056       Coll.: 1
Accesso: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Accesso   : 2013-01-07 12:16:52.815246252 +0100
Modifica  : 2013-01-07 12:16:52.815246252 +0100
Cambio    : 2013-01-07 12:16:52.815246252 +0100
Creazione: -
```

In questo caso notiamo vi è un unico file associato all'inode, infatti l'hard link, indicato come Coll. è semplicemente "1".

Linux non consente di creare hard link alle directory, questo per evitare problemi di ambiguità nel risalire la gerarchia di directory e per evitare che si possa includere parte di una gerarchia di directory all'interno di se stessa: cosa che creerebbe problemi ai programmi che esaminano ricorsivamente il contenuto di un file system, perché potrebbero non terminare mai il loro compito.

Per le directory il valore Coll. (hard link) indica il numero di sotto directory possedute, oltre al riferimento a sé stessa e alla directory superiore (. e..). Infatti il numero minimo per le directory è sempre 2. Possiamo vedere il valore di hard link anche mediante il comando ls -l, all'interno della colonna evidenziata:

Numero di hard link

```
roberto@roberto:~$ ls -l
totale 52
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Musica
-rw-r--r-- 1 root    root      14 gen  7 12:16 prova.txt
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Pubblici
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scaricati
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scrivania
drwxrwxr-x 2 roberto roberto 4096 dic 13 17:46 Ubuntu One
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Video
```

ESEMPIO 5 *Creare un hard link*

Abbiamo un file di nome prova.txt con inode pari a 669056, vogliamo crearne un collegamento di nome prova2, per fare questo usiamo il comando ln, seguito dal nome del file originale seguito dal nuovo hard link:

```
ln prova.txt prova
```

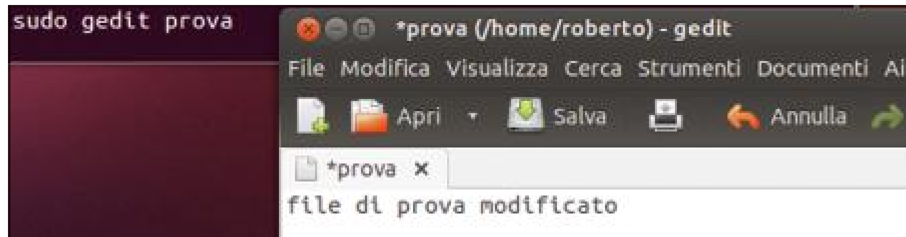
In questo caso prova sarà l'hard link del file prova.txt, eseguendo successivamente il comando ls -l notiamo come i file possiedano adesso 2 hard link: ▶

```
roberto@roberto: ~
roberto@roberto:~$ sudo ln prova.txt prova
roberto@roberto:~$ ls -l
totale 56
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Musica
-rw-r--r-- 2 root    root      14 gen  7 12:16 prova
-rw-r--r-- 2 root    root      14 gen  7 12:16 prova.txt
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Pubblici
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scaricati
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scrivania
drwxrwxr-x 2 roberto roberto 4096 dic 13 17:46 Ubuntu One
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Video
```

Mediante il comando ls -li scopriamo che i due file possiedono lo stesso numero di inode:

```
roberto@roberto:~$ ls -li
totale 56
789101 drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
669557 -rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
789103 drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
789077 drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
789102 drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Musica
669056 -rw-r--r-- 2 root    root      14 gen  7 12:16 prova
669056 -rw-r--r-- 2 root    root      14 gen  7 12:16 prova.txt
```

Il nuovo file punta allo stesso inode del precedente. Creando questo file non abbiamo occupato altro spazio sul disco, abbiamo semplicemente aggiunto un'altro riferimento allo stesso file. Se adesso modifichiamo il contenuto del file, attraverso un editor presente in Ubuntu (gedit), noteremo come la dimensione dei due file sarà la stessa:



Dopo aver salvato il file prova notiamo che la dimensione dei due file adesso è di 25 byte:

```
roberto@roberto:~$ ls -l
totale 60
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Musica
-rw-r--r-- 2 root    root      25 gen  7 16:44 prova
-rw-r--r-- 1 root    root      14 gen  7 16:44 prova~
-rw-r--r-- 2 root    root      25 gen  7 16:44 prova.txt
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Pubblici
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scaricati
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scrivania
drwxrwxr-x 2 roberto roberto 4096 dic 13 17:46 Ubuntu One
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Video
```

Possiamo notare che è stata salvata una versione precedente del file, il cui nome è lo stesso con aggiunto alla fine il simbolo della tilde (prova~).



Zoom su...

UTENTE E SUPER UTENTE

In Linux esiste un particolare utente chiamato super utente (super user o su), che possiede come UID (User ID) zero e come nome utente root. Tale utente possiede privilegi di amministratore e ha accesso al sistema senza nessuna restrizione. Nella maggior parte dei sistemi Linux, non viene usato il super utente per svolgere il compito di amministratore del sistema, bensì un utente diverso, per evitare rischi legati alla sicurezza. Per svolgere mansioni da amministratore possiamo aprire il Terminale e avviare una sessione come utente root. La distribuzione Ubuntu usa un sistema diverso per compiere operazioni amministrative, basato sull'utilizzo del comando sudo, attraverso il quale possiamo evitare di dover scrivere sempre la password.

Tuttavia in tal modo se un utente malintenzionato scopre la password di un utente abilitato all'utilizzo di sudo come root, può facilmente ottenere l'accesso totale al sistema. Inoltre se l'utente ha abilitato l'account root, utilizzando il comando su può ottenere i privilegi di amministrazione senza dover ripetere la password altre volte.

> Quando in un sistema alcuni compiti amministrativi sono condivisi fra diversi account, l'utilizzo di sudo evita di dover dare la password di root a più utenti.

Il comando sudo di Ubuntu, che tradotto significa super user do, consente di eseguire un comando come se si fosse un altro utente, in quanto effettua una specie di sostituzione, previa autorizzazione, tra l'utente attuale, cioè colui che esegue il comando sudo e l'utente target, cioè colui che esegue l'effettivo comando. Mentre con il comando su si cambia utente fino al termine della sessione del terminale, sudo assegna i privilegi dell'utente target al solo processo che viene con esso avviato.

Per eseguire dei comandi con privilegi d'amministrazione è sufficiente digitare sudo e successivamente il comando che si desidera eseguire come utente root, come nel seguente esempio:

```
sudo gedit prova.txt
```

L'utente target non deve essere necessariamente l'amministratore, ma può essere un qualsiasi utente del sistema. Per scegliere l'utente target, usare l'opzione -u:

```
sudo -u target comando
```

Dopo che abbiamo digitato il comando, il sistema chiederà la password dell'utente attuale e non la password dell'utente target, a meno che non si configuri sudo in modo diverso. La password viene chiesta la prima volta e memorizzata per un certo lasso di tempo, quindi è possibile usare il comando sudo più volte consecutive senza dover inserire ogni volta la password.

> Per motivi di sicurezza il Terminale non mostra nessun carattere di inserimento della password. A differenza delle richieste grafiche, nel terminale non vengono mostrati nemmeno gli asterischi di mascheramento. Digitare correttamente la password e premere Enter.

Con sudo l'amministratore del sistema può assegnare privilegi particolari a qualsiasi utente, definire quali comandi far eseguire e quali no, avere il log di tutte le operazioni effettuate e ricevere via email informazioni su tentativi di accesso non autorizzati. In Ubuntu, in modo predefinito, l'accesso come utente root è disabilitato, impedendo così l'utilizzo di su, ma permettendo comunque l'utilizzo di sudo. Questa scelta è dovuta a motivi di sicurezza. Tuttavia possiamo abilitare l'accesso come utente root assegnandogli una password con il seguente comando:

```
sudo passwd root
```

```
roberto@roberto:~$ sudo passwd root
Inserire nuova password UNIX:
Reinserire la nuova password UNIX:
passwd: password aggiornata correttamente
```

Adesso che la password per l'utente root è stata attivata, possiamo effettuare l'accesso come super user con il comando seguente:

```
sudo su
```

```
root@roberto: /home/roberto
roberto@roberto:~$ sudo passwd root
Inserire nuova password UNIX:
Reinserire la nuova password UNIX:
passwd: password aggiornata correttamente
roberto@roberto:~$ sudo su
root@roberto:/home/roberto#
```

Per disabilitare l'accesso come amministratore si può usare il seguente comando:

```
sudo passwd -l root
```

In questo modo l'accesso come utente root risulterà nuovamente bloccato.

File eseguibili e la variabile d'ambiente PATH

Affinché il sistema possa individuare i file eseguibili, senza costringere l'utente a conoscerne esattamente la posizione all'interno dell'albero delle directory, dobbiamo aggiungerne il percorso nella ◀ variabile d'ambiente ▶ PATH.



◀ **Variabile d'ambiente** È una particolare variabile presente nell'ambiente globale del S.O. accessibile dai processi tramite meccanismi di gestione dipendenti dal sistema operativo in uso. ▶

La variabile PATH contiene una stringa che rappresenta i percorsi in cui il sistema cercherà i file eseguibili. Ciascun percorso è separato dal successivo dai due punti (:). Le directory vengono esaminate nell'ordine con cui sono presenti nella stringa, per cui potremo avere più file eseguibili con lo stesso nome ma presenti in percorsi diversi, il file che verrà eseguito per primo sarà quello presente nella prima directory della sequenza. Per visualizzare il contenuto della variabile PATH usiamo il comando echo che effettua un output a video, seguito dal nome della variabile d'ambiente. seguente:

```
echo $PATH
```

```
roberto@roberto:~$ echo $PATH
/usr/lib/lightdm/lightdm:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin
:/bin:/usr/games:/usr/local/games
```

Per mostrare il contenuto della variabile, essa deve essere preceduta dal simbolo del dollaro ($).

Per modificare il contenuto della variabile d'ambiente PATH dobbiamo aggiungere la linea seguente:

```
export PATH=$PATH:/nuovo/percorso/da/aggiungere
```

al file.bashrc posizionato nella directory home dell'utente che intendi usare. Se il file non esiste dobbiamo crearlo facendo attenzione che il punto posto davanti al nome indica che il file è nascosto.

ESEMPIO 6

Generalmente l'avvio di un programma richiede l'indicazione del percorso, relativo o assoluto, per esempio per avviare l'applicazione che mostra la data di sistema (comando date) dobbiamo indicare espressamente l'eseguibile anteponendo il percorso assoluto o relativo:

```
roberto@roberto: ~
roberto@roberto:~$ cd /home
roberto@roberto:/home$ cd roberto
roberto@roberto:~$ /bin/date
dom 13 gen 2013, 11.25.16, CET
```

Tuttavia questo comando può essere eseguito in qualunque percorso ci troviamo in virtù del fatto che il percorso /bin è presente nella variabile d'ambiente PATH. Infatti proviamo a digitare il comando date da una directory qualsiasi del sistema e avremo il medesimo risultato:

```
roberto@roberto: /lib
roberto@roberto:/lib$ pwd
/lib
roberto@roberto:/lib$ date
dom 13 gen 2013, 11.34.22, CET
```

I comandi di rete ifconfig e ping

Il comando ifconfig consente di visualizzare a video le informazioni relative le interfaccie di rete presenti sulla macchina. Le schede di rete ethernet sono identificate dal codice eth seguito da un numero che parte da 0. Se per esempio abbiamo due interfaccie di rete, la prima sarà identificata da eth0 e l'altra da eth1.

> Quando un computer possiede più di un'interfaccia di rete oltre all'interfaccia di loopback, possiamo definirlo un multihomed host.

Per esempio:

```
roberto@roberto:/home$ ifconfig
eth0      Link encap:Ethernet  IndirizzoHW 08:00:27:e6:82:c3
          indirizzo inet:10.0.2.15  Bcast:10.0.2.255  Maschera:255.255.255.0
          indirizzo inet6: fe80::a00:27ff:fee6:82c3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1540 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1362 errors:0 dropped:0 overruns:0 carrier:0
          collisioni:0 txqueuelen:1000
          Byte RX:863678 (863.6 KB)  Byte TX:144828 (144.8 KB)

lo        Link encap:Loopback locale
          indirizzo inet:127.0.0.1  Maschera:255.0.0.0
          indirizzo inet6: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:54 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisioni:0 txqueuelen:0
          Byte RX:4362 (4.3 KB)  Byte TX:4362 (4.3 KB)
```

Con questo comando possiamo anche associare un indirizzo IP a una interfaccia, per esempio il comando seguente:

```
ifconfig eth0 192.168.1.10
```

Associa l'indirizzo all'interfaccia eth0.

Il comando ◀ ping ▶ consente di verificare la connessione tra due host.



◀ Ping Significa Packet internet grouper e misura il tempo, espresso in millisecondi, necessario per un pacchetto ICMP per raggiungere un altro dispositivo e ritornare all'origine. Viene usato in genere per verificare la presenza e la raggiungibilità di un host connesso in rete e misurarne le latenze di trasmissione di rete. ▶

La sintassi prevede 3 parametri principali:
- c seguito dal numero di volte con cui ripetere la prova;
- s seguito dal numero di byte che indicano la dimensione del pacchetto;
- l'indirizzo IP dell'host.

```
ping -c "n" -s "numero di volte" "ip destinazione"
```

Per esempio per verificare la connessione con un host di indirizzo 192.168.0.1 e per effettuare l'invio di 300 byte per pacchetto e per un numero massimo di 4 pacchetti digitiamo il seguente comando:

```
roberto@roberto:/home$ ping -c 4 -s 300 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 300(328) bytes of data.
308 bytes from 192.168.0.1: icmp_req=1 ttl=254 time=3.02 ms
308 bytes from 192.168.0.1: icmp_req=2 ttl=254 time=2.22 ms
308 bytes from 192.168.0.1: icmp_req=3 ttl=254 time=3.07 ms
308 bytes from 192.168.0.1: icmp_req=4 ttl=254 time=2.32 ms

--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.223/2.661/3.073/0.389 ms
```

Esercitazioni di Laboratorio 3

I COMANDI DI AMMINISTRAZIONE

L'amministrazione in ambiente shell Ubuntu

Uno degli aspetti più importanti nella gestione di un computer è l'amministrazione del sistema operativo. Le operazioni di manutenzione e di controllo delle periferiche, dei processi e delle memorie sono assai importanti in quanto garantiscono il corretto funzionamento della macchina. I comandi di amministrazione possono essere eseguiti solo come utente root, cioè l'utente che è amministratore di sistema e che possiede i permessi relativi, come abbiamo visto in precedenza. Pertanto dobbiamo impartire innanzi tutto il comando che serve per ottenere i privilegi di root:

```
sudo su
```

I processi costituiscono un fattore cruciale nell'economia di un sistema operativo: non accorgersi di alcuni problemi legati a un determinato processo può avere come conseguenza un drastico aumento delle risorse necessarie al suo funzionamento, con conseguente calo delle prestazioni. Innanzi tutto vediamo come terminare un processo, sappiamo che in Windows è possibile terminare un processo per mezzo della combinazione di tasti CTRL+ALT+CANC. In Linux digitiamo da terminale il comando seguente che elenca tutti i processi in esecuzione sul computer, disposti su 5 colonne:

```
ps x
```

```
roberto@roberto:~$ ps x
  PID TTY      STAT   TIME COMMAND
 1243 ?        Ssl    0:00 gnome-session --session=ubuntu
 1279 ?        S      0:00 dbus-launch --autolaunch=56cb3e64880a83db94b820c658ca04e7 --binary-syntax -
 1281 ?        Ss     0:00 //bin/dbus-daemon --fork --print-pid 5 --print-address 7 --session
 1283 ?        Ss     0:00 /usr/bin/ssh-agent /usr/bin/dbus-launch --exit-with-session gnome-session -
 1286 ?        S      0:00 /usr/bin/dbus-launch --exit-with-session gnome-session --session=ubuntu
 1287 ?        Ss     0:00 //bin/dbus-daemon --fork --print-pid 5 --print-address 7 --session
 1289 ?        Sl     0:00 /usr/lib/at-spi2-core/at-spi-bus-launcher
 1293 ?        S      0:00 /bin/dbus-daemon --config-file=/etc/at-spi2/accessibility.conf --nofork --p
 1297 ?        Sl     0:00 /usr/lib/at-spi2-core/at-spi2-registryd --use-gnome-session
 1304 ?        Sl     0:00 /usr/lib/gnome-settings-daemon/gnome-settings-daemon
 1305 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --start --components=gpg
 1366 ?        Sl     0:00 /usr/lib/gvfs/gvfsd
 1373 ?        Sl     0:00 /usr/lib/gvfs//gvfsd-fuse -f /run/user/roberto/gvfs
 1425 ?        Rl     0:22 compiz
 1430 ?        Sl     0:00 /usr/lib/dconf/dconf-service
 1434 ?        Sl     0:00 nautilus -n
 1435 ?        Sl     0:00 nm-applet
 1436 ?        Sl     0:00 /usr/lib/policykit-1-gnome/polkit-gnome-authentication-agent-1
 1437 ?        Sl     0:00 /usr/lib/gnome-settings-daemon/gnome-fallback-mount-helper
 1438 ?        Sl     0:00 bluetooth-applet
 1454 ?        Sl     0:00 /usr/lib/gvfs/gvfs-udisks2-volume-monitor
 1476 ?        Sl     0:00 /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
 1480 ?        Sl     0:00 /usr/lib/gvfs/gvfs-afc-volume-monitor
 1486 ?        S<l    0:00 /usr/bin/pulseaudio --start --log-target=syslog
 1495 ?        S      0:00 /usr/lib/pulseaudio/pulse/gconf-helper
 1497 ?        S      0:00 /usr/lib/i386-linux-gnu/gconf/gconfd-2
 1501 ?        Sl     0:00 /usr/lib/bamf/bamfdaemon
 1506 ?        Sl     0:00 /usr/lib/gvfs/gvfsd-trash --spawner :1.30 /org/gtk/gvfs/exec_spaw/0
```

Come possiamo notare nella prima colonna vi è il numero del processo e nell'ultima il nome del processo in esecuzione.

ESEMPIO 7 *Eliminare un processo*

In questo esempio vogliamo terminare il processo che mostra la barra del titolo delle finestre. Il processo che ci interessa è gtk-window-decorator:

```
1519 ?        Sl     0:00 /usr/bin/gtk-window-decorator
```

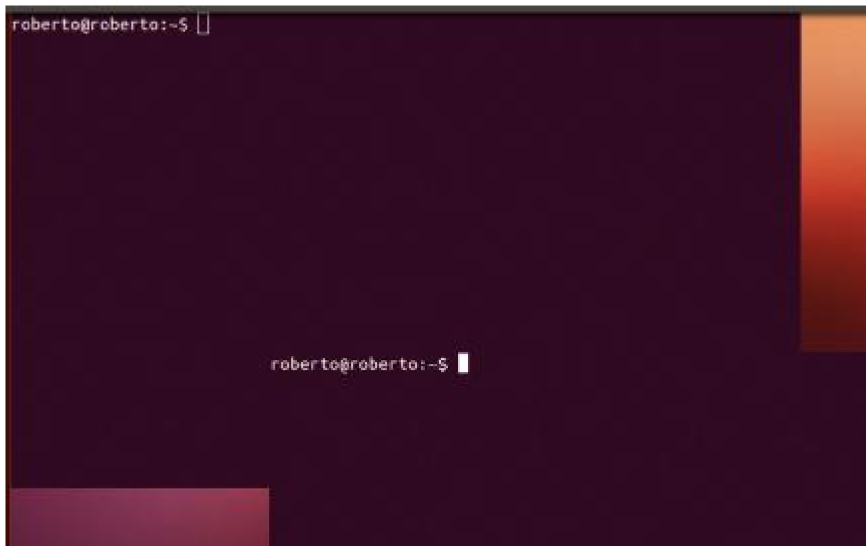
Dopo averlo individuato dobbiamo scrivere il comando kill, necessario per arrestare i processi, seguito dal numero del processo, in questo caso:

```
kill 1519
```

```
roberto@roberto:~$ kill 1519
```

Esiste anche un altro metodo per arrestare un processo, si tratta del comando killall seguito dal nome del programma. Un programma è in genere formato da diversi processi, quindi eliminare un programma significherà eliminarne tutti i processi.

Possiamo verificare che le finestre aperte successivamente saranno prive della barra superiore:





Prova adesso!

- Usare il Terminale Linux
- Gestire i processi

1. Prova a utilizzare il comando shutdown -h now, prima di tutto cerca di comprenderne l'uso mediante il manuale disponibile nel Terminale.
2. Utilizza il comando reboot, e verificane la differenza rispetto al comando suggerito nel punto 1.
3. Verifica il funzionamento del comando pstree, leggendone prima di tutto il manuale e successivamente testandolo nel Terminale.

La gestione delle directory

Innanzi tutto ricordiamo che Linux riconosce due tipi di percorsi dei file: assoluto o relativo. Come abbiamo visto nella lezione sul file system, il percorso assoluto indica in modo completo la destinazione da raggiungere, partendo dalla radice. La radice, indicata con lo slash (/) è la directory principale del sistema, ogni altro file o directory è contenuto al suo interno. Un percorso assoluto inizia quindi sempre con la barra e indicherà la posizione di un file all'interno della root, per esempio:

```
/dev/net
```

Il percorso relativo indica il percorso di un file partendo dalla posizione corrente e pertanto non inizia con lo slash (/). Per esempio il comando seguente:

```
../dev
```

indica un percorso che identifica la directory dev come "sorella" rispetto alla directory corrente, secondo lo schema ad albero delle directory, in quanto il simbolo del punto punto indica la directory superiore da cui accedere alla directory dev, che risulterà in tal modo "sorella" di quella in cui ci troviamo.

> Nei percorsi relativi possiamo usare due simboli che identificano la directory corrente (.) e la directory del livello superiore, o directory "madre" di quella corrente (..). Possiamo in tal modo identificare una directory o un file utilizzando i caratteri punto (.) e punto punto (..). Per conoscere la directory in cui ci troviamo (directory corrente) dobbiamo usare il comando pwd.

Appena accediamo al sistema tramite Terminale siamo posizionati nella directory /home dell'utente, per esempio nel nostro caso possiamo identificare la posizione con il comando pwd: ▶

```
roberto@roberto:~$ pwd
/home/roberto
```

Vediamo adesso il comando che consente di visualizzare il contenuto di una directory, si tratta di ls, abbreviazione di list. Permette ottenere un elenco sul terminale dei file e delle sottocartelle presenti nella directory corrente, per esempio:

```
roberto@roberto:~$ ls
Documenti         Immagini  Musica    Scaricati  Ubuntu One
examples.desktop  Modelli   Pubblici  Scrivania  Video
roberto@roberto:~$
```

Per poter visualizzare il contenuto della directory corrente possiamo anche usare il comando long listing format, che fornisce un elenco più completo e ricco di informazioni sui file e le directory contenute, la sintassi è: ls -l:

```
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Musica
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Pubblici
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scaricati
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scrivania
drwxrwxr-x 2 roberto roberto 4096 dic 13 17:46 Ubuntu One
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Video
```

> Avrete notato che le directory sono colorate di blu, per differenziarle dai file.

Per cambiare la directory corrente si usa il comando cd seguito dal percorso della directory da raggiungere. Proviamo per esempio a cambiare la directory corrente da /home/roberto alla directory /home/roberto/Documenti. Possiamo operare in due modi, usando un percorso assoluto oppure relativo. Nel primo caso il comando sarà il seguente:

```
cd /home/roberto/Documenti
```

Come possiamo notare il percorso della directory in cui ci troviamo adesso è il seguente: ~/Documenti, ricordando che il simbolo tilde (~) indica la directory home:

```
roberto@roberto: ~/Documenti
roberto@roberto:~$ cd /home/roberto/Documenti
roberto@roberto:~/Documenti$
```

Nel secondo caso (percorso relativo), il comando sarà il seguente, che tiene conto del fatto che la directory Documenti è una sottocartella rispetto alla directory corrente:

```
cd Documenti
```

Otteniamo sempre il medesimo effetto:

```
roberto@roberto: ~/Documenti
roberto@roberto:~$ cd Documenti
roberto@roberto:~/Documenti$
```

Per tornare alla directory /home possiamo digitare il comando cd senza parametri. Inoltre ricordiamo che è necessario lasciare uno spazio tra il comando cd e il successivo parametro.

ESEMPIO 8 *Usare il comando tree*

Esiste un comando che consente di visualizzare l'albero di tutte le directory, chiamato tree. Puoi scaricarlo e installarlo con il comando seguente:

```
sudo apt-get install tree
```

```
roberto@roberto:~$ sudo apt -get install tree
[sudo] password for roberto:
sudo: apt: command not found
roberto@roberto:~$ sudo-apt get install tree
sudo-apt: comando non trovato
roberto@roberto:~$ sudo apt-get install tree
Lettura elenco dei pacchetti... Fatto
Generazione albero delle dipendenze
Lettura informazioni sullo stato... Fatto
I seguenti pacchetti NUOVI saranno installati:
  tree
0 aggiornati, 1 installati, 0 da rimuovere e 226 non aggiornati.
È necessario scaricare 36,7 kB di archivi.
Dopo quest'operazione, verranno occupati 112 kB di spazio su disco.
Scaricamento di:1 http://it.archive.ubuntu.com/ubuntu/ quantal/universe tree i38
6 1.6.0-1 [36,7 kB]
Recuperati 36,7 kB in 0s (117 kB/s)
Selezionato il pacchetto tree non precedentemente selezionato.
(Lettura del database... 153043 file e directory attualmente installati.)
Estrazione di tree (da .../archives/tree_1.6.0-1_i386.deb)...
Elaborazione dei trigger per man-db...
Configurazione di tree (1.6.0-1)...
```

A questo punto possiamo utilizzare il comando tree per visualizzare l'albero delle directory della directory in cui ci troviamo, otteniamo:

```
roberto@roberto:~$ tree
.
├── Documenti
├── examples.desktop
├── Immagini
├── Modelli
├── Musica
├── Pubblici
├── Scaricati
├── Scrivania
├── Ubuntu One
│   └── Shared With Me -> /home/roberto/.local/share/ubuntuone/shares
└── Video

10 directories, 1 file
```

Creare, cancellare e copiare directory

I comandi che consentono di creare, cancellare e copiare le directory sono i seguenti:

- mkdir: crea una nuova directory;
- rmdir: cancella una directory, solo se vuota;
- rm: cancella una directory o un file;
- cp: copia una directory o un file.

Creare una nuova directory significa assegnare un nome a una cartella nel percorso desiderato, in grado di contenere file oppure altre cartelle. Copiare una directory significa copiarne anche tutto il contenuto. Cancellare una directory con rm significa eliminarla per sempre, mentre con rmdir possiamo eliminare solo una directory non vuota.
Vediamo innanzi tutto come creare una nuova directory con il comando mkdir:

```
mkdir prova
```

In questo caso abbiamo creato la directory prova all'interno del percorso corrente. L'esempio che segue mostra come creare una directory in un percorso diverso indicandolo attraverso un percorso assoluto. Innanzi tutto posizioniamoci nella directory home, possiamo vederne il percorso attraverso il comando tree:

```
roberto@roberto:/home$ tree
.
└── roberto
    ├── Documenti
    ├── examples.desktop
    ├── Immagini
    ├── Modelli
    ├── Musica
    ├── Pubblici
    ├── Scaricati
    ├── Scrivania
    ├── Ubuntu One
    │   └── Shared With Me -> /home/roberto/.local/share/ubuntuone/shares
    └── Video
```

Desideriamo creare una directory come sotto cartella di Musica, di nome rock. Pertanto il suo percorso assoluto sarà il seguente: /home/roberto/Musica/rock.
Digitiamo il comando e quindi osserviamo se la directory è stata effettivamente creata: ▶
Come possiamo notare la directory è stata creata.

```
roberto@roberto: /home
roberto@roberto:/home$ mkdir /home/roberto/Musica/rock
roberto@roberto:/home$ tree
.
└── roberto
    ├── Documenti
    ├── examples.desktop
    ├── Immagini
    ├── Modelli
    ├── Musica
    │   └── rock
    ├── Pubblici
    ├── Scaricati
    ├── Scrivania
    ├── Ubuntu One
    │   └── Shared With Me -> /home/roberto/.local/share/ubuntuone/shares
    └── Video
```



Prova adesso!

• Usare i comandi mkdir, cd, ls

1. Crea una directory di nome soul nella directory Scaricati, con un percorso relativo.
2. Adesso prova a posizionarti nella directory appena creata sopra e visualizzane il contenuto.

Per copiare una cartella si utilizza il comando cp -r seguito dal percorso della directory da copiare e dal percorso di dove destinarla. Per esempio vogliamo copiare la cartella rock nella directory /home.
Il comando necessario, che utilizza percorsi assoluti, è il seguente:

```
sudo cp -r /home/roberto/Musica/rock /home
```

Il parametro -r è necessario per copiare le directory, mentre invece vedremo che per copiare solo i file non dovrà essere utilizzato.

Vediamo l'esecuzione del comando indicato sopra, come possiamo notare in seguito al comando tree possiamo verificare che la cartella sia stata copiata: ▼

Come avrete notato è stato usato il comando sudo poiché i comandi di copia e cancellazione possono essere effettuati solo dall'utente root (super user) in possesso di privilegi maggiori.

```
roberto@roberto: /home
roberto@roberto:/home$ sudo cp -r /home/roberto/Musica/rock /home
roberto@roberto:/home$ tree
.
├── roberto
│   ├── Documenti
│   ├── examples.desktop
│   ├── Immagini
│   ├── Modelli
│   ├── Musica
│   │   └── rock
│   ├── Pubblici
│   ├── Scaricati
│   ├── Scrivania
│   ├── Ubuntu One
│   │   └── Shared With Me -> /home/roberto/.local/share/ubuntuone/shares
│   └── Video
└── rock
```

Per eliminare le cartelle possiamo usare due comandi, rmdir che elimina la directory solo se è vuota e rm -r che elimina la directory indipendentemente dal suo contenuto. Entrambi i comandi prevedono che il percorso della directory dal eliminare venga scritto di seguito dopo lo spazio. Proviamo a eliminare la directory rock creata nel percorso /home:

```
roberto@roberto:/home$ rmdir /home/roberto/Musica/rock
roberto@roberto:/home$ tree
.
├── roberto
│   ├── Documenti
│   ├── examples.desktop
│   ├── Immagini
│   ├── Modelli
│   ├── Musica
│   ├── Pubblici
│   ├── Scaricati
│   ├── Scrivania
│   ├── Ubuntu One
│   │   └── Shared With Me -> /home/roberto/.local/share/ubuntuone/shares
│   └── Video
└── rock
```

Come possiamo notare la directory è stata eliminata.

Possiamo anche eliminare più cartelle indicandone i nomi di seguito separati dallo spazio, per esempio per eliminare due directory digitiamo i due percorsi subito dopo il comando di cancellazione:

```
rm -r /home/prova /home/roberto/Musica/rock
```

ESEMPIO 9 *Creare una directory e un file e cancellarli*

Adesso proviamo a creare una directory, scrivere al suo interno un file e cancellarla con il comando rm -r che elimina la directory e il suo contenuto. Prima di tutto creiamo la directory prova nel percorso /home verifichiamone la creazione con tree:

```
roberto@roberto:/home$ sudo mkdir prova
roberto@roberto:/home$ tree
.
├── prova
```

Adesso posizioniamoci nella directory prova:

```
roberto@roberto:/home$ cd prova
roberto@roberto:/home/prova$
```

Quindi creiamo un file di nome pippo attraverso l'utile comando touch che crea un file vuoto secondo una tecnica che viene chiamata on the fly (al volo):

```
roberto@roberto:/home/prova$ sudo touch pippo
roberto@roberto:/home/prova$ ls -l
totale 0
-rw-r--r-- 1 root root 0 gen 12 19:14 pippo
```

Adesso usciamo dalla directory, tornando alla directory madre (/home) con il comando cd..; da qui digitiamo il comando di cancellazione e verifichiamo l'avvenuta eliminazione con il comando tree:

```
roberto@roberto:/home/prova$ cd ..
roberto@roberto:/home$ sudo rm -r /home/prova
roberto@roberto:/home$ tree
.
├── roberto
│   ├── Documenti
│   ├── examples.desktop
│   ├── Immagini
│   ├── Modelli
│   ├── Musica
│   ├── Pubblici
│   ├── Scaricati
│   ├── Scrivania
│   ├── Ubuntu One
│   │   └── Shared With Me -> /home/roberto/.local/share/ubuntuone/shares
│   └── Video
└── rock
```

Come possiamo notare attraverso il comando digitato sopra abbiamo eliminato sia la directory prova che il file contenuto in essa.

Le directory possono essere spostate mediante il comando mv seguito dal percorso della directory da spostare e il percorso in cui collocarla. Per esempio per spostare la directory /home/roberto/Musica/rock in /home/roberto impostiamo il seguente comando:

```
sudo mv /home/roberto/Musica/rock /home/roberto
```



Prova adesso!

- Utilizzare il Terminale
- Applicare i comandi per le directory

1. Posizionati nella directory /home e crea una sottodirectory di nome esempio.
2. Posizionati nella directory esempio.
3. Torna alla directory /home.
4. Crea una sotto cartella della directory /home chiamata prova.
5. Posizionati adesso nella directory prova.
6. Posizionati nella directory esempio usando un percorso relativo.
7. Crea un file nella directory prova.
8. Copia la directory prova sotto alla directory esempio.



Zoom su...

I CARATTERI JOLLY

In Linux esistono dei caratteri speciali che vengono interpretati dal sistema per poter selezionare più nomi di file in un unico comando. Tali caratteri vengono chiamati caratteri jolly o wildcard.

Il più importante tra questi è l'asterisco (*) che viene interpretato come qualsiasi digitazione di caratteri. Per esempio per poter visualizzare tutti i file contenuti in una directory inizianti con la lettera "r" indichiamo il comando:

```
ls r*
```

In tal modo informiamo il sistema che il nome del file interessato possiede la lettera "r" come primo carattere seguito da un numero imprecisato di altri simboli. Oppure volendo copiare da una directory a un'altra tutti i file che iniziano con la parola "eser" (quindi esercizio, oppure esercitazione, ma anche esercitiamoci ecc.) digitiamo:

```
cp /home/roberto/eser* /home
```

In questo caso tutti i file che sono trovati nella directory /home/roberto vengono copiati nella directory /home.
Oppure per cancellare tutti i files dalla mia directory basterà utilizzare il comando:

```
rm *
```

In tal modo i files che hanno per nome qualsiasi digitazione di caratteri (*) saranno eliminati, quindi tutti quelli presenti nella directory.
Un altro carattere jolly è rappresentato dal punto interrogativo (?). Si tratta di un carattere speciale che viene interpretato come un qualsiasi singolo carattere. Per esempio con il comando seguente:

```
ls pr???
```

Si ottengono tutti i file che sono lunghi al massimo 5 caratteri e iniziano con la stringa "pr", quindi per esempio "prova".
Caratteri speciali sono, infine, le parentesi quadre [ ]: queste consentono di specificare un determinato insieme di caratteri da ricercare nel nome del file. Se per esempio vogliamo elencare i file che iniziano per "prova" e finiscono con una A,B o C (per esempio il file provaA) allora basterà usare il comando:

```
ls prova[ABC]
```

Il comando cat

Questo comando serve sia per concatenare dei file che per visualizzare a video il contenuto dei file di testo. Il codice che segue visualizza il contenuto del file lettera:

```
cat lettera
```

Il comando che segue mostra il contenuto di due file (lettera1 e lettera2) concatenandoli sullo schermo:

```
cat lettera1 lettera2
```

File e permessi

Come abbiamo visto in precedenza, il comando ls -l mostra a video un elenco di file e directory assai dettagliato che mostra in particolare il tipo di file, i permessi e l'appartenenza all'utente e al gruppo. Per esempio nel seguente elenco notiamo i permessi per tutti i file indicati:

```
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-rw-r-- 1 roberto roberto    0 gen 13 21:16 esempio
-rw-r--r-- 1 roberto roberto 8445 dic 13 17:27 examples.desktop
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Immagini
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Modelli
drwxr-xr-x 2 roberto roberto 4096 gen 12 19:05 Musica
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Pubblici
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scaricati
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Scrivania
drwxrwxr-x 2 roberto roberto 4096 dic 13 17:46 Ubuntu One
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Video
```

Colonna dei permessi relativi ai file e alle directory

Colonne utente e gruppo di appartenenza

Colonna della dimensione in Byte

Colonna della data e ora di creazione del file

Come possiamo notare nella prima colonna vi sono i permessi degli utenti sul relativo file (lettura, scrittura, esecuzione), nella seconda e terza colonna sono indicati il gruppo di appartenenza e l'utente di appartenenza del file, quindi la dimensione in byte, infine la data e l'ora della creazione del file o della directory indicata.

Vediamo in dettaglio come sono indicati i permessi: il primo carattere a sinistra indica un attributo specifico del file, per esempio nel caso il file rappresenti una directory comparirà il carattere d. I successivi nove caratteri rappresentano i permessi e sono divisi in tre terne:

Terna	Valore mostrato dal comando ls-l	Utente
prima terna	vwx --- ---	owner (proprietario)
seconda terna	--- rwx ---	group (gruppo di utenti)
terza terna	--- --- vwx	other (altri)

Sono presenti tre diverse restrizioni di accesso: ▶

Simbolo	Permesso
r	read (lettura)
w	write (scrittura)
x	execute (esecuzione)

Nel caso dei file abbiamo che:
- r (lettura) consente di aprire un file per visualizzarne il contenuto;
- w (scrittura) consente di sovrascrivere o aggiungere dati a un file;
- x (esecuzione) consente di eseguire un file (nel caso si tratti di un file eseguibile).

Nel caso delle directory abbiamo che:
- r (lettura) consente la visualizzazione del contenuto della directory, con il comando ls;
- w (scrittura) consente la creazione o l'eliminazione di file all'interno della directory;
- x (esecuzione) consente di accedere alla directory (per esempio con il comando cd) anche nel caso non se ne possa visualizzare il contenuto.

Applicare il permesso di scrittura a una directory consente la cancellazione dei file contenuti in essa anche a utenti che non possiedono i permessi di scrittura su tali file.

La modifica dei permessi dei file e delle directory avviene per mezzo del comando chmod la cui sintassi è la seguente:

```
chmod [OPZIONI] permessi nomefile
```

Ci sono due metodi per modificare i permessi: mediante i numeri o mediante i caratteri. In generale sconsigliamo la modifica dei permessi dei file di sistema, alcuni file possiedono permessi assai restrittivi con lo scopo preciso di evitare accessi non autorizzati e conseguenti problemi di sicurezza. Per esempio, il file /etc/shadow, che contiene le password utente, non ha impostato alcun permesso per gli utenti.

La tabella a fianco mostra il significato dei caratteri che possono essere indicati nelle OPZIONI del comando chmod: ▶

Opzione	Definizione
u	proprietario
g	gruppo
o	altri
x	esecuzione
w	scrittura
r	lettura
+	aggiungi permesso
-	annulla permesso
=	imposta permesso

Per esempio per consentire la lettura, scrittura e l'esecuzione al proprietario di un file usiamo il seguente comando:

```
chmod o+rwx esempio
```

ESEMPIO 10 *Assegnare e rimuovere permessi*

In questo esempio ci accingiamo a creare quattro nuovi file e assegnare e rimuovere da essi i relativi permessi. Innanzi tutto creiamo i quattro file, rispettivamente esempio1, esempio2, esempio3, esempio4 con il comando seguente:

```
roberto@roberto:~$ touch esempio1 esempio2 esempio3 esempio4
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-rw-rwx 1 roberto roberto    0 gen 13 21:16 esempio
-rw-rw-r-- 1 roberto roberto    0 gen 13 22:01 esempio1
-rw-rw-r-- 1 roberto roberto    0 gen 13 22:01 esempio2
-rw-rw-r-- 1 roberto roberto    0 gen 13 22:01 esempio3
-rw-rw-r-- 1 roberto roberto    0 gen 13 22:01 esempio4
```

Togliamo il permesso di lettura agli altri utenti (other) del primo file (esempio1), mediante il comando:

```
chmod o-r esempio1
```

Otteniamo: ▶

```
roberto@roberto:~$ chmod o-r esempio1
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-rw-rwx 1 roberto roberto    0 gen 13 21:16 esempio
-rw-rw---- 1 roberto roberto    0 gen 13 22:01 esempio1
```

Al termine di ogni comando verifichiamo sempre le modifiche con il comando ls-l.

Adesso proviamo ad aggiungere agli altri utenti la scrittura e lettura al file esempio2:

```
chmod o+wx esempio2
```

Otteniamo: ▶

```
roberto@roberto:~$ chmod o+rw esempio2
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-rw-rwx 1 roberto roberto    0 gen 13 21:16 esempio
-rw-rw---- 1 roberto roberto    0 gen 13 22:01 esempio1
-rw-rw-rw- 1 roberto roberto    0 gen 13 22:01 esempio2
```

Ora neghiamo il permesso di lettura e scrittura al gruppo sul file esempio3 ed esempio4:

```
chmod g-rw esempio3 esempio4
```

Otteniamo: ▶

```
roberto@roberto:~$ chmod g-rw esempio3 esempio4
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rw-rw-rwx 1 roberto roberto    0 gen 13 21:16 esempio
-rw-rw---- 1 roberto roberto    0 gen 13 22:01 esempio1
-rw-rw-rw- 1 roberto roberto    0 gen 13 22:01 esempio2
-rw----r-- 1 roberto roberto    0 gen 13 22:01 esempio3
-rw----r-- 1 roberto roberto    0 gen 13 22:01 esempio4
```

Adesso proviamo ad aggiungere i permessi di lettura, scrittura ed esecuzione a tutti gli utenti e a tutti e quattro i file:

```
chmod ugo+rwx esempio*
```

Come avete visto in questo ultimo esempio è stato usato il carattere jolly asterisco (*) che consente di raggruppare tutti i file che iniziano per esempio.

```
roberto@roberto:~$ chmod ugo+rwx esempio*
roberto@roberto:~$ ls -l
totale 48
drwxr-xr-x 2 roberto roberto 4096 dic 13 17:40 Documenti
-rwxrwxrwx 1 roberto roberto    0 gen 13 21:16 esempio
-rwxrwxrwx 1 roberto roberto    0 gen 13 22:01 esempio1
-rwxrwxrwx 1 roberto roberto    0 gen 13 22:01 esempio2
-rwxrwxrwx 1 roberto roberto    0 gen 13 22:01 esempio3
-rwxrwxrwx 1 roberto roberto    0 gen 13 22:01 esempio4
```



Prova adesso!

• Usare i comandi sui permessi

1. Crea due file chiamandoli esempio1 e esempio2.
2. Assegna al primo il permesso di lettura all'utente proprietario e al secondo solo l'esecuzione all'utente group.
3. Elimina i due file creati sopra.
4. Crea altri due file chiamandoli file1 e file2.
5. Assegna a file1 tutti i permessi a tutti gli utenti e togli al file2 tutti i permessi a tutti gli utenti.

Modificare la proprietà di un file

Per modificare la proprietà di un file o di una directory possiamo utilizzare il comando chown (change owner). Tuttavia data l'estrema potenza del comando può essere eseguito solo dall'utente root. La sintassi specifica è:

```
chown nuovo_proprietario nome_file
```

Per esempio per assegnare la proprietà del file esempio1 all'utente mario dobbiamo usare il comando:

```
sudo chown mario esempio1
```

La gestione degli utenti

Gli utenti Linux sono le persone che interagiscono con il sistema. Possiamo definirne più di uno in quanto abbiamo setto in precedenza che Linux è un S.O. multiutente; la definizione degli utenti si rende necessaria in quanto il sistema deve riconoscere l'utente al momento del suo accesso in maniera da poter attuare per esso le politiche di accesso alle varie parti del sistema in base al suo profilo.
L'insieme degli utenti che possono accedere al sistema è memorizzato all'interno del sistema stesso e ciascun utente è identificato univocamente da un valore numerico chiamato UID (User IDentifier) e a questo sono associati uno username (il nome dell'utente) e una password (parola d'ordine). Al momento dell'accesso, il sistema richiede all'utente di inserire il proprio username e per verificare che si tratti proprio di lui, richiede anche la relativa password, che solo l'utente in questione dovrebbe conoscere. Se l'utente viene riconosciuto, in base allo username e alla password inseriti, questi viene fatto accedere all'interfaccia utente del sistema. Vi è una distinzione tra utenti standard e amministratore del sistema, quest'ultimo è colui che lo gestisce e lo configura, e viene anche chiamato super user e possiede come nome root e UID uguale a 0. Per tutti gli altri utenti il sistema attua un meccanismo di controllo di accesso a tutte le varie risorse.
Gli utenti Linux possiedono una directory personale chiamata cartella home, una propria configurazione di sistema e altri parametri personalizzati. Più utenti possono essere connessi al sistema ed è possibile controllare quali utenti sono collegati e quali sono le operazioni e i processi a essi associati in ogni momento attraverso il comando w, l'amministrare del computer può in tal modo tenere sotto controllo gli accessi al sistema. Vediamo l'output del comando w:

```
roberto@roberto:~$ w
 10:01:53 up 0 min,  2 users,  load average: 0,60, 0,18, 0,06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
roberto  tty7     :0               10:01   46.00s  2.20s  0.14s gnome-session -
roberto  pts/1    :0               10:01    0.00s  0.03s  0.00s w
```

Per gestire gli utenti e i gruppi di utenti l'amministratore può utilizzare diversi comandi, riassunti nella tabella seguente:

Comando	Significato
useradd [opzioni] nomeutente	Aggiunge un utente, è possibile tramite le varie opzioni disponibili modificare tutte le impostazioni di default di inserimento utenti
userdel [opzioni] nomeutente	Elimina un'utente, da sottolineare che la sua home con il suo contenuto non viene cancellata
groupadd [opzioni] nomegruppo	Aggiunge un gruppo
passwd [nomeutente]	Modifica la password di un'utente
chsh [opzioni]	Cambia il tipo si shell disponibile al login di un'utente

La redirezione

Tutti i comandi Linux possiedono un input e un output. Alcuni di essi necessitano di esplicitare l'input oppure l'output mentre altri li possiedono in modo implicito, per esempio un comando di copia possiede come input e output il file da copiare e la sua destinazione mentre invece il comando ls possiede come output lo schermo, come output implicito.
La maggior parte dei processi avviati da comandi Linux scrivono sullo standard output cioè lo schermo e ricevono l'input da tastiera. Oltre agli input e output classici (schermo e tastiera) esiste anche lo standard error, dove i processi scrivono i loro messaggi d'errore, e che coincide con lo schermo.

ESEMPIO 11 *Redirezione del comando cat*

In questo esempio scopriremo che il comando cat possiede un input e un output di tipo implicito, si tratta dello standard input (tastiera) e dello standard output (video). Adesso digitiamo il comando cat senza alcun parametro, in modo da attivare l'input e l'output impliciti nel comando. Come possiamo notare il cursore si attende che scriviamo qualcosa. Dopo aver digitato del testo premiamo Invio, notiamo che il testo viene copiato sullo standard output, cioè lo schermo. Per uscire dal comando premiamo CTRL D che termina la sequenza di input.

```
roberto@roberto:~$ cat
sal a tutti sto scrivendo sullo stdinput e appena premerò invio invierò sullo st
doutput
sal a tutti sto scrivendo sullo stdinput e appena premerò invio invierò sullo st
doutput
roberto@roberto:~$
```

La redirezione avviene attraverso quattro simboli principali:
- > serve per redirigere l'output dei comandi;
- < serve per redirigere l'input dei comandi;
- >> server per redirigere l'output accodando i dati a un file;
- | serve per redirigere entrambi, connettendo l'output di un comando con l'input del successivo.

La redirezione dell'output

Come indicato sopra il simbolo > viene usato per redirigere l'output di un comando. Per esempio vogliamo redirigere l'output del comando cat per poter scrivere da tastiera un elenco di nomi da collocare nel file elenco. Per fare questo andiamo a redirigere solo l'output del comando cat, in quanto l'input standard di tale comando è proprio la tastiera, ricordando di premere CTRL D per terminare l'immissione: ▶

```
roberto@roberto:~$ cat > elenco
Mario
Sandra
Luca
Chiara
roberto@roberto:~$ cat elenco
Mario
Sandra
Luca
Chiara
```

Il comando cat ha quindi ricevuto l'input dalla testiera e lo ha rediretto attraverso il file elenco. Abbiamo poi visualizzato il contenuto del file elenco con il comando cat elenco.

L'espressione >> aggiunge lo standard output in coda a un file, pertanto se vogliamo per esempio aggiungere altri elementi al file elenco visto nell'esempio precedente digitiamo il comando:

```
cat >> elenco
```

Dopo aver scritto un nuovo elenco e salvato CTRL D esso verrà aggiunto al file esistente.



Prova adesso!

- Usare la redirezione dell'output

1. Utilizzando i comandi di redirezione visualizza il contenuto della directory /home all'interno del file casa.
2. Crea un file chiamandolo elenco2 contenente un elenco di frutti.
3. Mostra il contenuto del file elenco2 all'interno del file elenco3 usando la redirezione dell'output.

La redirezione dell'input

Il carattere < indica l'intenzione di redirigere l'input di un comando. Per comprenderne il funzionamento useremo un comando particolare, chiamato sort, che consente di ordinare un elenco di valori, riga per riga, secondo i caratteri ascii contenuti. Innanzi tutto dobbiamo dire che il comando sort possiede input e output impliciti, l'input è la tastiera mentre l'output è rappresentato dallo schermo. Proviamo a eseguire il comando sort, dopo aver digitato una serie di nomi, riga per riga, terminiamo l'inserimento con CTRL D: otterremo a video l'elenco ordinato:

```
roberto@roberto:~$ sort
mario
barbara
martino
luca
anna
vincenzo
sara
riccardoanna
barbara
luca
mario
martino
riccardo
sara
vincenzo
roberto@roberto:~$
```

Immissione CTRL D

ESEMPIO 12 *Ordinamento di un elenco*

In questo esempio innanzi tutto scriviamo un elenco di nomi in un file di nome elenco attraverso il comando cat: ▶

```
roberto@roberto:~$ cat > elenco
vincenzo
gino
luca
anna
maria
gino
```

Adesso utilizziamo il comando sort per ordinare l'elenco presente nel file. Per fare questo dobbiamo utilizzare il simbolo < che consente di passare il file elenco come input per il comando sort. Otteniamo un elenco ordinato a video: ▶

```
roberto@roberto:~$ sort <elenco
anna
gino
gino
luca
maria
vincenzo
```

Infine per ottenere un elenco ordinato in un altro file non dovremo fare altro che redirigere anche l'output del comando sort verso un altro file: ▶

```
roberto@roberto:~$ sort < elenco > elenco_ord
roberto@roberto:~$ cat elenco_ord
anna
gino
gino
luca
maria
vincenzo
```

Il collegamento tra input e output successivo con la pipe

Passare i dati da un comando a un altro significa passare i risultati in output dell'esecuzione di un comando all'input del comando successivo. Per fare questo è necessario usare il carattere ascii chiamato pipe (|). Per esempio per inviare alla stampante (indicata dalla parola lpr) il contenuto di una directory digitiamo il seguente comando:

```
ls | lpr
```

dove lpr indica la stampante e il simbolo pipe mette in contatto i due comandi: in tal modo l'output del comando ls, quindi il contenuto della directory, viene inviato al comando lpr che lo riceverà come input e provvederà a stamparlo. Le pipe in una riga possono essere anche più di una, come per esempio:

```
sort elenco | cat -n | lpr
```

In questo caso il file elenco viene ordinato, quindi passato al comando cat -n che lo numera e lo passa al comando successivo che lo stamperà.

- Alcuni esempi di redirezione dell'output:

```
ls – l > listato
```

Trasferisce l'output del comando ls –l sul file listato, pertanto a video non viene visualizzato nulla in quanto l'output è stato collocato nel file listato. Il file se esistente viene sovrascritto.

```
pwd > dir_corrente
```

La directory corrente viene scritta nel file di nome dir_corrente.

```
whoami > nome
```

Il file nome contiene ora lo username corrente.

```
whoami >> nome
```

Il comando accoda il nome utente al contenuto del file nome.

- Alcuni esempi di redirezione dell'input:

```
sort < elenco
```

Il file chiamato elenco viene ordinato e mostrato sullo standard output che per il comando sort è lo schermo.

Se non vogliamo avere un output a video di un comando possiamo effettuare una redirezione nulla, indicando come parametro di redirezione il percorso del file speciale /dev/null, una sorta di cestino.

ESEMPIO 13 *Vediamo chi è connesso*

Il comando che consente di conoscere gli utenti connessi al sistema è who. Esso possiede come standard output il video. Il comando mostra tutte le console, anche quelle testuali, in cui attualmente è autenticato un utente. Per ottenerne una lista ordinata possiamo usare la pipe nel modo seguente:

```
who | sort
```

La figura seguente mostra il comando who digitato senza redirezione e il comando con l'ordinamento degli utenti: ▶

```
roberto@roberto:~$ who
roberto  tty7         2013-01-14 17:50 (:0)
roberto  pts/0        2013-01-14 17:52 (:0)
roberto@roberto:~$ who | sort
roberto  pts/0        2013-01-14 17:52 (:0)
roberto  tty7         2013-01-14 17:50 (:0)
```

Per scoprire quanti utenti sono attualmente connessi al tuo computer, digitiamo:

```
who | wc -l
```

Il risultato è il seguente: ▶

```
roberto@roberto:~$ who | wc -l
2
```

Un altro utilizzo della **pipe** è legato al comando more. Il comando more possiede come input implicito un file di testo e un output implicito che è lo schermo. Il file di testo viene mostrato a video a pagine. Pertanto risulta assai utile utilizzarlo insieme al comando | per visualizzare a pagine un file particolarmente grande, agevolandone in tal modo la lettura: ▶

```
roberto@roberto:~$ cat elenco | more
mario
luca
alfonso
gianluca
domizia
carmela
assunta
marina
alessia
vincenzo
martina
marco
monica
chiara
elena
sofia
viola
anna
matteo
marinella
sandro
--Ancora--
```



Prova adesso!

- Usare la redirezione di input e output

1. Crea due file contenenti un elenco di nome e chiamali elenco1 ed elenco2.
2. Visualizza i due file ordinandoli in un unico elenco nel file elenco3.

Verifichiamo le conoscenze

Esercizi a scelta multipla

1 Indica quali permessi possiede il seguente file:
`d---rw-r--`

a) È una directory, lettura e scrittura per il gruppo e lettura per gli altri
b) Lettura per il proprietario, lettura e scrittura per il gruppo, lettura per gli altri
c) Lettura per il proprietario, lettura e scrittura per il gruppo, esecuzione per gli altri
d) È una directory, lettura per il gruppo, scrittura e lettura per gli altri

2 Indica cosa effettua il seguente comando:
sudo passwd -l root

a) Modifica la pqssword dell'utente superuser
b) Elimina la password del superuser
c) Crea un nuovo superuser
d) Disabilita l'accesso come amministratore

3 Scegli quale tra i seguenti è il nome dell'utente superuser:

a) Administrator
b) admin
c) root
d) su

4 Indica il comando che mostra l'elenco dei file i relativi inode:

a) ls -il
b) ls
c) cat
d) ls-li

5 Quale riga tra le seguenti mostra la sintassi completa di un comando?

a) nome_comando man
b) man nome_comand
c) nome_comando -help
d) nome_comando --help

6 Il comando gparted consente di:

a) programma che mostra tutti gli utenti
b) programma che mostra il file system montato
c) programma che mostra le partizioni
d) programma che crea e modifica le partizioni

7 Il comando touch:

a) crea un nuovo file vuoto
b) mostra il file system montato
c) mostra le partizioni
d) mostra i processi attivi

8 Come si chiama il file system di Linux?

a) ntfs
b) ext

c) si
d) text

9 Linux rappresenta:

a) il S.O.
b) il kernel del S.O.
c) gli script del S.O.
d) La shell del S.O.

10 Indica il significato del comando seguente in relazione al file elenco:

chmod o+rwx elenco
a) consente la lettura, scrittura e l'esecuzione al proprietario del file
b) consente la lettura, scrittura e l'esecuzione agli altri utenti
c) consente la lettura, scrittura e la visualizzazione agli altri utenti
d) consente la lettura, scrittura e l'esecuzione agli altri utenti

11 Le directory sono comunque file?

a) falso
b) vero, infatti hanno il permesso ugo
c) vero
d) vero, infatti possiedono il permesso d

12 Quale comando consente di entrare in ambiente SuperUser?

a) sudo root
b) non si deve fare niente, Linux è sempre in super user
c) su
d) sudo du

13 Quale comando consente la modifica dell'indirizzo IP: 192.168.1.110?

a) ifconfig eth0 192.168.1.110
b) ipconfig eth0 192.168.1.110
c) ipconfig 192.168.1.110
d) ipconfig 192.168.1.110

14 I comandi shell si scrivono:

a) in minuscolo, Linux è case sensitive
b) in maiuscolo, Linux è case sensitive
c) dipende dal comando, comunque Linux non è case sensitive
d) in maiuscolo, comunque Linux non è case sensitive

15 Indica il comando necessario per spostare il controllo alla directory "sorella" di quella in cui ci troviamo:

a) cd ../altra_dir
b) cd /altra_dir
c) cd ./altra_dir
d) cd /../ altra_dir

16 Indica come si riconosce l'ambiente Superuser?

a) Dal prompt che termina col simbolo dollaro ($)
b) Dal prompt chetermina col simbolo chiocciola (@)
c) Dal prompt che termina coi due punti (:)
d) Dal prompt che termina col simbolo hash (#)

17 **Quale directory rappresenta il desktop, per un utente che si chiama mario?**

a) /mario/home/Scrivania
b) /home/Desktop
c) /dev/mario/Desktop
d) /home/mario/Scrivania

18 **Quale comando installa un package?**

a) get-install
b) apt
c) apt install
d) apt-get install

19 **Per cancellare una directory non vuota usiamo il comando:**

a) md
b) rmdir
c) cp
d) rm -r

20 **Quale comando tra i seguenti copia tutti i file che iniziano per "p" alla directory del livello superiore?**

a) cp p* //
b) copy p*.* ..
c) cp p* ..
d) cp p*.*

21 **Quale comando cancella tutti i file della radice?**

a) rm ../*
b) rm /*
c) rm -r /*
d) rm .*

22 **Quale comando permette di creare un file di testo da tastiera, chiamato elenco?**

a) cat elenco
b) cat > elenco
c) cat >> elenco
d) cat < elenco



VERSIONE
SCARICABILE
EBOOK

e-ISBN 978-88-203-5839-6

www.hoepli.it

Ulrico Hoepli Editore S.p.A.
via Hoepli, 5 - 20121 Milano
e-mail hoepli@hoepli.it